ReliaBle EmBedded Systems

Das Yocto Projekt - Eine ausführliche Übersicht

Robert Berger - Reliable Embedded Systems e.U.
Consulting Training Engineering
https://www.ReliableEmbeddedSystems.com
robert.berger@ReliableEmbeddedSystems.com

Kurs Ziele

Aufbauend auf Ihr Wissen betreffend Embedded Linux, dem Linux Kernel und weiteren Komponenten eines Linux Systems möchten wir Ihnen das Wesentliche zur Nutzung des Yocto Projekts vermitteln. Wir werden sehen, wie ein BSP/Framework-Betreuer das Yocto Projekt verwenden würde, sowie Entwickler, die vielleicht nicht einmal wissen wollen, dass sie es verwenden.

Beschreibung

Diese viertägige Schulung kombiniert Theorie mit Hands-on Übungen, um das Yocto Projekt vorzustellen. Sie beantwortet häufig gestellte Fragen wie:

- /> Was ist das Yocto Projekt und was ist der OE/Yocto workflow?
- /> Was ist BitBake?
- /> Was sind Layers?
- /> Ist es wirklich notwendig, für jedes Linux Projekt eine andere Version der Toolchain/Bibliotheken/Pakete zu verwenden, und dann auch noch bei jedem Projekt einem anderen Workflow zu folgen?
- /> Können Sie sicherstellen, dass die Entwicklungsumgebung für alle Entwickler/Lieferanten identisch ist und dass man in 10+ Jahren immer noch die gleichen Builds wie heute reproduzieren kann?
- /> Kann das YP dabei helfen herausfinden, unter welchen Softwarelizenzen die von Ihnen verwendeten Pakete lizenziert sind?
- /> ... und vieles mehr

Hands-On-Sessions werden auf dem Host, in einem Docker-Container und auf einer bestimmten Zielhardware (z.B. i.mx6 [1]) ausgeführt. Sie erhalten ein brandneues Hardware-Kit [1] damit Sie in der Lage sind auch nach der Schulung das Gelernte zu wiederholen bzw. mit neuen Dingen zu experimentieren. Nach dem Training können Sie einen auf Ubuntu basierenden Docker-Container mit allen Abhängigkeiten schon vorinstalliert sowie sämtliche Beispiele herunterladen um mit dem Kursmaterial in Ihren eigenen Labors zu arbeiten. Bitte beachten Sie, dass dies kein Einführungskurs in Embedded Linux ist, wie der, den wir hier [2] anbieten. Sie sollten bereits wissen, wie Embedded Linux funktioniert und wie der Linux-Kernel bzw. Kerneltreiber konfiguriert/erstellt werden.

Erforderliche Vorkenntnisse

- /> Grundlegende Vertrautheit mit der Verwendung eines Linux-Systems (z.B. Ubuntu) als Endbenutzer im user space

- /> Grundlegende Vertrautheit mit einer Befehlszeilen-Shell
- /> Grundkenntnisse von User/Kernel-Space Programmierung unter Linux
- /> Mittelmäßige C-Programmierkenntnisse
- /> Sie sollten *"Embedded Linux - Von Systemarchitektur zu Echtzeit (5 Tage)"* [2] oder *"Einführung in Embedded Linux in Theorie und Praxis - ein Intensivkurs (3 Tage)"* [3] besucht haben. (dringend empfohlen!)
  …es sei denn Sie sind mit dem darin vorgetragenen Material schon sehr vertraut.
  Um dem Training folgen zu können sollten Sie Erfahrung mit Embedded Linux mitbringen (wie man den Kernel baut, userspace, root file-system, tools).
- /> Es könnte auch hilfreich sein wenn Sie *"Embedded Linux Kernel Interna und Gerätetreiber (5 Tage)"* [4]" besucht haben, aber das ist keine erforderliche Vorkenntnis. Es reicht wenn Sie wissen wie man den Linux Kernel, Kerneltreiber in/out of tree baut und was ein fdt ist um dem Kernel Kapitel folgen zu können.
- /> Die oben erwähnten Schulungen können auch on-site und on-line angeboten werden. Hier [5] können Sie Angebote dazu einholen.

Wer sollte teilnehmen?

Sie verwenden bereits Linux für Ihre Projekte und haben wahrscheinlich schon von dem Yocto Projekt gehört, haben es aber nicht gewagt, es näher zu betrachten oder hatten Schwierigkeiten damit. Sie wissen nicht, ob und wie Ihr täglicher Workflow mit dem YP zu realisieren ist und finden das YP im Allgemeinen ziemlich kompliziert. Warum brauchen wir all das, obwohl (angeblich) alles bisher viel einfacher war? Nach dem Training sollten Sie entscheiden können, ob Sie das YP brauchen oder nicht. Der Workshop richtet sich an Software-, Entwicklungs-, Systemingenieure, Tester, Administratoren, Ingenieure und andere an dem YP Interessierte, die bereits fundierte Kenntnisse von Embedded Linux haben.

Kursinhalt

Tag 1

Einführung

- /> Geschichte von Unix/Linux | Lizenzen | Standards | Die Unix Philosophie

Besonderheiten von "Embedded"

- /> Linux auf dem Desktop im Vergleich zu Linux auf einem Eingebetteten System | Cross-/Native Toolchains | schon vorgebaute Toolchains | Build-Systeme | C-Bibliotheken | …

Eval Board

- /> Wie booted Linux auf einem PC und auf dem Eval Board? | Sehen wir wie es booted

Yocto Einführung

- /> Was ist Yocto? | Was ist das YP? (Features/Herausforderungen) | Einige Werkzeuge unter dem projektübergreifenden YP (Poky | BitBake | OE-Core | Meta Daten) | Warum sollte man das YP verwenden?

So werden Sie Teil unserer Yocto Project-Community (optional)

/> Erste Schritte | So erreichen Sie uns | Sich beteiligen (Participate) | Beitragen (Contribute) | Soziale Medien | Veranstaltungen

Entwicklungsumgebung

/> Was wird für einen YP-Build benötigt? | Layer Versionen/Abhängigkeiten | Hostabhängigkeiten | Konfigurationsdateien | Host-/Kernel-Abhängigkeiten

Der YP Workflow

/> Einleitung

/> Konfiguration (Benutzer | Meta Daten | Maschine | Regeln für die Distribution)

/> Eigenschaften (Features) | Versionierung von Rezepten | Layers

/> Quellen

/> Quellen fetchen | Patchen | Konfigurieren/Kompilieren/Installieren | `SSTATE` | Pseudo | `recipetool` | Beispiele für Rezepte | `PACKAGECONFIG` | Pakete und deren Inhalt | Output-Analyse/Packaging | Erzeugen von Images | Erzeugen von SDKs | Tasks

/> Anpassen von Images (Intro | `local.conf` | `IMAGE_FEATURES` | benutzerdefinierte `.bb` files | benutzerdefinierte Paketgruppen)

Tag 2

BitBake

/> Geschichte

/> Syntax (Ersetzen des Variablennamens durch den Inhalt [Variable Expansion] | Variablenzuweisung [Variable Assignment] | Vor-/Anhängen [Pre-/Append] | Entfernen [Removal] | Bedingte [Conditional] Syntax - `OVERRIDES`)

/> BitBake Debugging (Debug Level | Rezepte/Images/Packagegroups finden | BitBake Umgebung/-Tasks/Logging | bauen/spezifischen Task erzwingen | `cleansstate` | `stamp` für ungültig erklären | `devshell` | Abhängigkeits-Explorer | alle BitBake Instanzen töten | grafische Wrapper für BitBake)

/> Aufräumen (um Speicherplatz zu gewinnen | damit neu gebaut werden muss)

Layer

/> Intro | `bitbake-layers` tool | dynamische Layer

BSP

/> Intro | Systementwicklungs-Workflow | BSP Entwicklungsanleitung (`bsp-tool` - auf die aktuelle Poky Version portiert | Erstellung eines BSPs | Patches, die noch nicht Mainline sind)

Kernel

/> Intro | Systementwicklungs-Workflow | Kernel-Entwicklungsanleitung (`defconfig` | `defconfig` + Konfigurationsfragment | in tree kmod | out of tree kmod | "klassischer" fdt | fdt mit `devicetree.bbclass` | ...)

Tag 3

Software Entwicklungskit

/> Einführung | Cross-Entwicklungs Toolchain | Sysroot | BBCLASSEXTEND | Multilib | Der QEMU-Emulator | Eclipse Yocto Plug-in (veraltet) | User Space Werkzeuge | Installieren von SDKs und Toolchains

/» Cross-Toolchains/SDKs

../ Erstellen eines Cross-Toolchain-Installers

../ Verwenden des Standard-SDK (Cross-Toolchain+Makefile/Autotools/Autotools lib + App | Rezepte)

../ Erstellen/Verwenden eines Erweiterbaren (Extensible) SDK

Bibliotheken (Libraries)

/> Namenskonvention für .so files | Was soll wo hin (rootfs/SDK)? | Wie kann man ein autotooled .a mit Poky verwenden?

Eclipse (veraltet/optional)

/> Einführung | Anwendungsentwicklungs-Workflow

/> Mit Eclipse arbeiten (rootfs features | kernel + fdt | rootfs | JDK installieren | Eclipse installieren | Eclipse konfigurieren | Das Yocto Plugin installieren | Das Yocto Plugin konfigurieren | Ein "Autotooled Projekt" generieren | Die Cross-Toolchains konfigurieren | Das Projekt bauen)

Tag 4

User Space Fehlersuche/-behebung

/> Intro | gdb | gdb Remote-Debugging | (gdb Remote) Debugging mit Eclipse | (remote) Ausführen mit Eclipse

Profiling/Tracing

/> Intro | perf | gprof | gcov | strace | ftrace | systemtap | LTTng + Eclipse (Datenvisualisierung) | top | powertop | powerdebug

Paketverwaltung

/> Softwareaktualisierungen (SW Update) im Vergleich zu "Paket Management" | Mit Paketen arbeiten | IPK | Einen package feed erstellen | Installieren eines Pakets mit opkg auf dem Zielsystem

Lizenzierung

/> Einführung | eine benutzerdefinierte Lizenz hinzufügen | eine "kommerzielle" Lizenz hinzufügen | Firmware Lizenzen | Open-Source-Lizenz Bestimmungen

Devtool

/> Einführung | Ein Rezept hinzufügen/bauen/installieren | Einen Layer erstellen/hinzufügen | Finish | Ein Rezept Ändern/Aktualisieren | Bauen/Ausführen | Ein Image bauen

Technische Voraussetzungen für die Teilnahme an einer Remote-/Online-Schulung

- (ship-it/Web/Host/Zielhardware/Telefon) E-Mail-Adresse, um Login Informationen zu erhalten.
- (Web) Screen Sharing/Audio/Video/Whiteboard/Chat/Fragen&Antworten:
  `https://www.bigmarker.com` erfordert dies: `https://rlbl.me/bm-req`.
- Backup: (Web)/Audio (Telefon) Telefonkonferenz:
  `https://www.turbobridge.com/international.html`
- (Host/Zielhardware) Shell: Port 22 nicht blockiert:
  so etwas wie: `ssh <user>@vlabx.dyndns.org`
- Backup: (Host/Zielhardware) Shell über Browser: Port 443 nicht blockiert:
  so etwas wie: `https://vlabx.dyndns.org`

Remote-/Online-Buchungsoptionen

Alle Schulungsunterlagen sind in englischer Sprache, der Vortrag kann entweder auf Englisch oder Deutsch erfolgen, wie Sie es wünschen, weltweit.

Während des Trainings bieten wir folgendes an	**remote/online öffentlich** 1 Teilnehmer von einer Firma	**remote/online privat** ab 3 Teilnehmer[2] in einer Buchung z. B. von einer Firma
	GTR[1] ab 3 Teilnehmer mit Schulungsleiter	
(ship-it) Arbeitsbuch	auf Englisch/pdf - Download Link, gedruckt auf Anfrage✓ behält der Schulungsteilnehmer✓	
(ship-it) Beispiele	Download Link + am Host installiert✓ behält der Schulungsteilnehmer✓	
(target) Kit [1]	wird dem Teilnehmer geliefert✓ um nach dem Training mit Trainingsmaterial experimentieren zu können✓ behält der Schulungsteilnehmer✓	
(ship-it) Build Umgebung	Download-/Installationsanweisungen + auf dem Host installiert✓ behält der Schulungsteilnehmer✓	
(Host/Target) Fern-Zugriff	✓	
(host) Host Maschine	Laptop/VM/Container - gehostet✓	
(target) Board	gehosted✓	
(target) Board Power over IP	gehosted✓	
(target) Board Zugriff auf Konsole	gehosted✓	
(target) Board booted über tftp/nfs	gehosted✓	
(web) Screen Sharing	gehosted✓	
(web) Audio/Video	gehosted✓	
(web) Whiteboard	gehosted✓	
(web) Chat	gehosted✓	
(web) F&A[3]	gehosted✓	
backup: (web) Audio	Telefonkonferenz - weltweite Einwahlnummern✓	
(host/target) Shell	gehosted - ssh (port 22) - ssh client✓	
Backup: (host/target) Shell über https	gehosted - https (port 443) - Browser✓	
benutzerdefinierter Inhalt		Der Inhalt kann an Ihre Bedürfnisse angepasst werden[4]✓
benutzerdefinierte Probleme		unternehmens-/projektspezifische Probleme können diskutiert & vielfach gelöst werden ✓
große Flexibilität		Beratung inkludiert[5]✓

[1] guaranteed to run - findet statt
[2] wenn Sie 3 Plätze und mehr buchen gibt es Rabatt
[3] Fragen und Antworten
[4] Kapitel meiner vorhandenen Schulungen können hinzugefügt/entfernt werden
[5] während des Trainings

Ähnliche Kurse

/> *"Embedded Linux - Von Systemarchitektur zu Echtzeit (5 Tage)"*
https://rlbl.me/elisa
https://rlbl.me/elisa-de-pdf

/> *"Auffrischungskurs Embedded Linux & Einführung in das Yocto Project (5 Tage)"*
https://rlbl.me/intely
https://rlbl.me/intely-r-de-pdf

/> *"Embedded Linux Kernel Interna und Gerätetreiber (5 Tage)"*
https://rlbl.me/ldd
https://rlbl.me/ldd-de-pdf

/> *"Das Yocto Project - Eine ausführliche Übersicht (4 Tage)"*
https://rlbl.me/yocto
https://rlbl.me/yocto-r-de-pdf

/> *"Einführung in Embedded Linux & Echtzeit, das Yocto Project aus der Vogelperspektive (4 Tage)"*
https://rlbl.me/entirety
https://rlbl.me/entirety-de-pdf

/> *"Embedded Linux und Anschließen von Hardware (4 Tage) - kommt bald"*
https://rlbl.me/elisha

/> *"Linux Gerätereiber kurz und bündig (4 Tage) - kommt bald"*
https://rlbl.me/coldd

/> *"Einführung in Embedded Linux in Theorie und Praxis - ein Intensivkurs (3 Tage)"*
https://rlbl.me/elin
https://rlbl.me/elin-de-pdf

/> *"(Embedded) Linux Fehlersuche-/behebung (3 Tage)"*
https://rlbl.me/lindeb
https://rlbl.me/lindeb-r-de-pdf

/> *"FreeRTOS in Theorie und Praxis (3 Tage)"*
https://rlbl.me/freertos
https://rlbl.me/freertos-r-de-pdf

/> *"Das Zephyr Project - ein Überblick (3 Tage) - kommt bald"*
https://rlbl.me/zephyr

/> *"Das Yocto-Projekt für Fortgeschrittene - Lizenzierung (1 Tag)"*
https://rlbl.me/yopalic
https://rlbl.me/yopalic-r-de-pdf

Verweise

[1] *"Zielhardware"*
https://rlbl.me/hw

[2] *"Embedded Linux - Von Systemarchitektur zu Echtzeit (5 Tage)"*
https://rlbl.me/elisa

[3] *"Einführung in Embedded Linux in Theorie und Praxis - ein Intensivkurs (3 Tage)"*
https://rlbl.me/elin

[4] *"Embedded Linux Kernel Interna und Gerätetreiber (5 Tage)"*
https://rlbl.me/ldd

[5] *"Reliable Embedded Systems: Kontakt"*
https://rlbl.me/contact

[6] *"Vortragsmethoden für Schulungen"*
https://rlbl.me/delivery

Trainer

Robert Berger hat seit 1993 Praxis- und Führungserfahrung in der Industrie bei Design und Entwicklung von Software für "Eingebettete Systeme" mit und ohne harten Echtzeit Anforderungen. Er arbeitet seit dem Beginn des 21. Jahrhunderts mit Linux auch im Desktop/Server Bereich, aber hauptsächlich mit "Eingebetteten Systemen" (Automotive, Industrial Control, Robotics, Telecom, Consumer Electronics,...). Man trifft ihn regelmäßig bei internationalen Events wie z.B. "Embedded World", "Embedded Software Engineering Kongress", "Embedded Systems Conference", "Embedded Linux Conference" und "Yocto Project Summit" als Experten und Vortragenden. Seine Spezialitäten sind vorwiegend Schulungen aber auch Beratungen (auf Deutsch oder Englisch) weltweit und zwar von kleinsten Echtzeit Systemen (FreeRTOS) bis hin zu Systemen mit mehreren Prozessoren/Kernen und Embedded Linux (User-, Kernel-Space, Gerätetreiber, Hardware Interfacing, Debugging, Multi-core, Yocto Projekt,...) mit Schwerpunkt Free- und Open Source Software. Robert ist ein Weltenbummler. Er ist CEO & Embedded Software Evangelist bei Reliable Embedded Systems e.U. (St. Barbara/Österreich) und wenn er keine Kundenbesuche macht ist er bei seiner Familie in Athen/Griechenland.

Vielen Dank für Ihr Interesse!

Für Anfragen schicken Sie bitte eine E-mail an:
training@ReliableEmbeddedSystems.com