



Certificate of Advanced Studies

CAS Embedded Secure Linux

Internet of Things, Android auf Consumer Devices, flexible Embedded Lösungen, allen ist eines gemeinsam: Ohne Open Source-Tools und -Plattformen, im speziellen Linux, wären sie kaum möglich. Der stark wachsende Markt verlangt von Entwickler*innen in diesen Gebieten fundierte Kenntnisse der aktuellen Werkzeuge und durch die hohe Vernetzung einen starken Fokus zum Thema Sicherheit. Das CAS Embedded Secure Linux vermittelt Ihnen hierzu alles nötige Rüstzeug.



bfh.ch/cas-ebstx

Inhaltsverzeichnis

1	Umfeld	4
2	Zielpublikum	4
3	Ausbildungsziele	4
4	Voraussetzungen	4
5	Durchführungsort	4
6	Kompetenzprofil	5
7	Kursübersicht	6
8	Kursbeschreibungen	6
	8.1 Neuerungen von C++11 bis C++20	6
	8.2 Linux, Build-Environment, Tools und Chains, Testing	7
	8.3 Embedded Android und Kommunikation	7
	8.4 Security	8
	8.5 Integrationsprojekt	8
9	Kompetenznachweis	9
10	Ergänzende Lehrmittel	9
11	Dozierende	10
12	Organisation	10

Stand: 04.07.2022

1 Umfeld

Eingebettete Systeme übernehmen heute in allen möglichen Bereichen unserer Umwelt Überwachungs-, Steuerungs- und Regelungsfunktionen. Die Basis vieler Embedded-Lösungen ist Linux. Daneben stehen heute weitere Betriebssysteme wie Embedded Android für den Bau von IoT-Devices zur Verfügung. Bei der durchdringenden Vernetzung aller Systemen müssen auch die Sicherheitsaspekt zwingend beachtet werden.

Das CAS Embedded Secure Linux vermittelt Ihnen Konzepte und Umsetzungsmöglichkeiten anhand von Linux und Android, dabei werden auch die neusten Sprachmittel von C++ und deren Einsatz in Embedded Systemen vorgestellt. Sie lernen, was zu beachten ist, um ein sicheres Embedded-System mit dem besten Nutzen für den Anwender zu entwickeln.

2 Zielpublikum

Personen, die Soft- und Hardware entwickeln, sowie Ingenieur*innen, die im Bereich Embedded Systems tätig sind.

3 Ausbildungsziele

- Sie erwerben fundierte Kenntnisse zu Konzeption, Planung und Design von sicheren Embedded-Systemen.
- Sie setzen Linux und Android als Entwicklungsplattform und Technologie professionell ein, um Embedded-Systeme zu realisieren.
- Sie können evaluieren, in welchem Kontext Android als Teil einer Embedded-Lösung Vorteile bringt.
- Sie lernen die neuen Sprachelemente von C++11 bis C++20 und deren Einsatz in Embedded Systemen kennen.

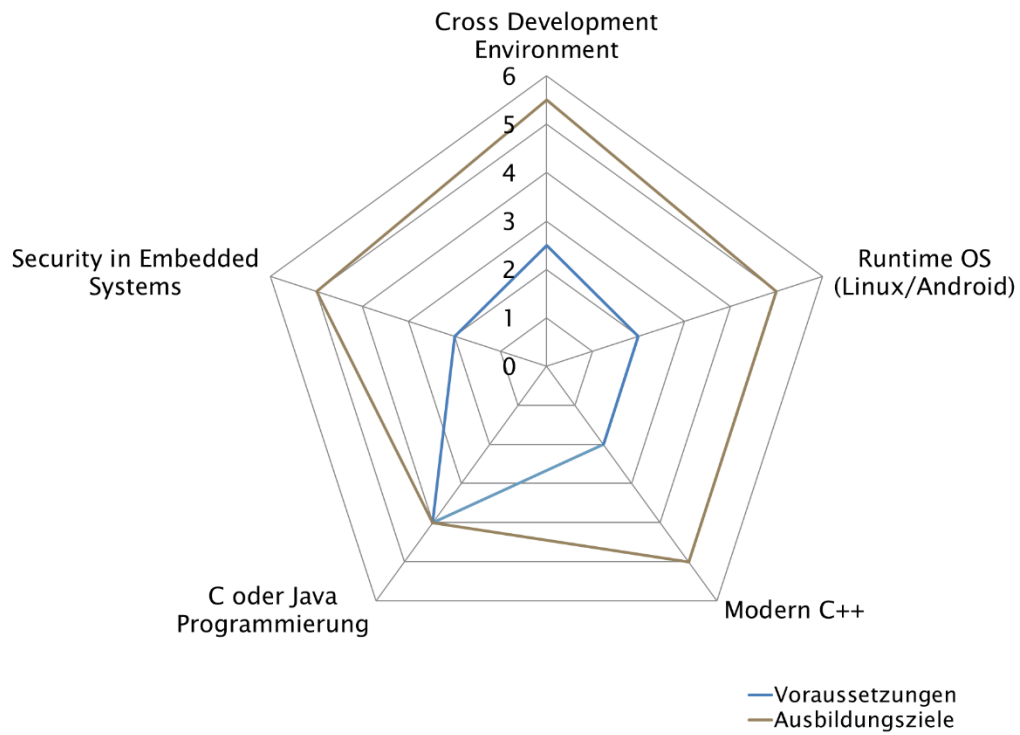
4 Voraussetzungen

Gute Kenntnisse der Sprache C/C++ und/oder Java sowie Erfahrung im Umgang mit Linux oder anderen Embedded Entwicklungsumgebungen sind zwingend erforderlich. Vorkenntnisse im Bereich Realtime Systeme und der Internet-Security sind von Vorteil.

5 Durchführungsort

Berner Fachhochschule, EIT Burgdorf, Jlcoweg 1, 3400 Burgdorf,
Telefon: +41 34 426 68 84, E-Mail: ivo.oesch@bfh.ch

6 Kompetenzprofil



Kompetenzstufen

1. Kenntnisse/Wissen
2. Verstehen
3. Anwenden
4. Analyse
5. Synthese
6. Beurteilung

7 Kursübersicht

Kurs / Lehreinheit	Lektionen	Stunden	Dozierende
Neuerungen von C++11 bis C++20	24	~20	Ivo Oesch
Linux, Build-Environment, Tools und Chains, Testing	40		Martin Aebersold Dominik Nufer
Embedded Android und Kommunikation	16	~16	Martin Aebersold
Security	28	~20	Dominik Nufer
Projektarbeit	20	~20	Alle
Organisation und Meilensteine der Integrationsprojekte	8		Ivo Oesch (Rolf Lanz)
Integrationsprojekt		~100	
Total	136	~176	

Das CAS umfasst insgesamt 12 ECTS-Credits. Für die einzelnen Kurse ist entsprechend Zeit für Selbststudium, Prüfungsvorbereitung etc. einzurechnen.

8 Kursbeschreibungen

Nachfolgend sind die einzelnen Kurse dieses Studienganges beschrieben.

Der Begriff Kurs schliesst alle Veranstaltungstypen ein, es ist ein zusammenfassender Begriff für verschiedene Veranstaltungstypen wie Vorlesung, Lehrveranstaltung, Fallstudie, Living Case, Fach, Studienreise, Semesterarbeiten usw.

8.1 Neuerungen von C++11 bis C++20

Lernziele	Die Teilnehmenden haben detaillierte Kenntnisse der Neuerungen der Sprachstandards C++11 bis C++20 und sind in der Lage diese produktiv einzusetzen.
Themen und Inhalte	<ul style="list-style-type: none"> – Async Library (Threads und Synchronisation) – Coroutinen – Weitere Neuerungen der Sprachstandards C++11 bis C++20 – Qt Library: GUI sowie Signal und Slot Konzept – Angewandte Beispiele
Lehrmittel	– Skript, das alle wesentlichen Lerninhalte umfasst

8.2 Linux, Build-Environment, Tools und Chains, Testing

Lernziele	Die Teilnehmenden verstehen die Funktionsweise aller beteiligten Toolchains und sind in der Lage, ein Embedded System mit Hilfe der GNU-Tools zu erstellen, zu konfigurieren und zu testen.
Themen und Inhalte	<ul style="list-style-type: none"> – Einführung in die GNU Tools (gcc, make, etc.) – Aufbau, Installation und Unterhalt der Cross-Compile Umgebung – Aufbau und Unterhalt eines Embedded Root-Filesystems – Konfiguration des Bootloaders (U-Boot) und des Kernels – Anpassen des Device-Tree für eine Zielhardware. – Linux I/O aus dem user space (GPIO, ADC, PWM, I2C, SPI) – Das Frambuffer-Device, GUI-Toolkits, QT5 – Design und Implementierung von einfachen Gerätetreibern – Realtime: Preemption Patch und OpenAMPFramework – Building an Embedded Linux System: Yokto, Buildroot, Homemade – Testsystem: Fuego Test System, LAVA, Testing und Simulation
Lehrmittel	<ul style="list-style-type: none"> – Kommentiertes Folienset – Literaturempfehlungen Nr. 7, 8

8.3 Embedded Android und Kommunikation

Lernziele	Die Teilnehmenden verstehen den Aufbau von Embedded Android, können dieses produktiv einsetzen und sind insbesondere in der Lage zu entscheiden, bei welchen Projekten es sinnvoll ist, ein Embedded Android einzusetzen.
Themen und Inhalte	<ul style="list-style-type: none"> – Gesamtarchitektur und Einführung in das Open Source Project (AOSP) – Übersicht SDK, NDK, Emulator und Debugger – Kompilieren der kompletten AOSP Quellen für ein Zielgerät – I/O Programmierung mit Java. (GPIO, ADC, PWM) – Aufbau und Funktion der JNI-Schnittstelle – Portierung und Anpassen von Android für ein Zielgerät. – Kommunikation: Einbindung von BLE-Komponenten und Einsatz von MQTT
Lehrmittel	<ul style="list-style-type: none"> – Kommentiertes Folienset – Literaturempfehlung Nr. 6

8.4 Security

Lernziele	<p>Die Teilnehmenden verfügen über das nötige Fachwissen um die häufigsten Sicherheitslücken zu erkennen und wissen diese zu unterbinden und gar nicht erst aufkommen zu lassen.</p> <p>Die Teilnehmenden kennen das Prinzip von Public Keys und beherrschen den Umgang mit digitalen Zertifikaten.</p>
Themen und Inhalte	<ul style="list-style-type: none"> – Mathematische Grundlagen für Security – Top ten security risks – Public Key Infrastruktur, Umgang mit digitalen Zertifikaten. – Bootloader: Secure Boot, Arm TrustZone Technology, Open Portable Trusted Execution Environment (OP-TEE) – Safe and Secure Updating: Deployment, OTA Updates, RAUC, SecDevOps
Lehrmittel	<ul style="list-style-type: none"> – Kommentiertes Folienset

8.5 Integrationsprojekt

Lernziele	<p>Das Integrationsprojekt kann vertraulich behandelt werden. Massgebend für die Rahmenbedingungen ist das Studienreglement. Die Vertraulichkeit darf den didaktischen Rahmen nicht behindern: Präsentationen und Diskussionen über das gewählte Thema müssen im Rahmen der Klasse möglich sein.</p>
Themen und Inhalte	<p>Im Integrationsprojekt bearbeiten die Teilnehmenden ein Embedded-Projekt aus ihrer Firma oder Ihrem beruflichen Umfeld. Mit dem gewählten Thema vertiefen die Studierenden die im Studium erlernten Methoden und lernen diese in der Praxis anzuwenden.</p>
Ablauf	<p>Das Integrationsprojekt umfasst ca. 100 (max. 120) Arbeitsstunden. Es ist eine Gruppen- oder Einzelarbeit und beinhaltet die folgenden Meilensteine (siehe auch Zeitplan):</p> <ol style="list-style-type: none"> 1. In der Firma ein Thema suchen und finden, sowie eine Ansprech-/Betreuungsperson in der Firma definieren. 2. Erstellen einer Projektskizze (siehe unten). Anschliessend Eingabe an die CAS-Leitung 3. Individuelle Kurzpräsentation (10') und Diskussion (10') des Themas an der Schule vor einem Dozierenden-Gremium. 4. Eventuell Überarbeitung der Projektskizze gemäss Feedback an der Präsentation. 5. Die BFH ordnet eine*n Expert*in für die Begleitung des Semesterprojekts zu. 6. Durchführung der Arbeit in eigener Terminplanung. 7. 2-3 Meetings mit Expert*in. <ul style="list-style-type: none"> – Projektskizze besprechen / Kick-Off – Ev. bei Bedarf: Zwischenreview / Beratung – Schlusspräsentation vor Expert*innen- und Dozierenden-Gremium. 8. Abgabe des Berichtes an Expert*in und an die Schule (auf Studienplattform, per Email oder auf Wunsch in Papierform). 9. Beurteilung durch Expert*in.

Ablauf	<p>Die Projektskizze umfasst eine ein- bis maximal zweiseitige Aufgabenstellung mit folgenden Teilen:</p> <ol style="list-style-type: none"> 1. Titel 2. Umfeld 3. Problemstellung 4. Lösungsansatz (Vorgehen, Methoden) 5. Name und Kontaktadressen der Gruppenmitglieder, und der Ansprech-/Betreuungsperson in der Firma <p>Gruppenarbeiten sind, wo immer möglich, erwünscht und je nach Rahmenbedingungen meist von Vorteil. Der nominelle Aufwand liegt bei ca. 100h, kann je nach Vorbereitungsphase und Komplexität der Aufgabe aber auch leicht höher sein.</p>
Ergebnis und Bewertung	Der Bericht umfasst ca. 20 Seiten. Der Source Code ist sofern für die Projektbeurteilung notwendig, als Anhang mitzuliefern.

9 Kompetenznachweis

Für die Anrechnung der 12 ECTS-Credits ist das erfolgreiche Bestehen der Qualifikationsnachweise (Prüfungen, Projektarbeiten) erforderlich, gemäss folgender Aufstellung:

Kompetenznachweis	Gewicht	Art der Qualifikation	Erfolgsquote Studierende
Neuerungen von C++11 bis C++20	1	Projektarbeit & Prüfung	0 - 100 %
Linux, Build-Environment, Tools und Chains, Testing	2	Projektarbeit & Prüfung	0 - 100 %
Embedded Android und Kommunikation	1	Prüfung	0 - 100 %
Security	1	Prüfung	0 - 100 %
Integrationsprojekt	5	Einzel- oder Gruppenarbeit	0 - 100 %
Gesamtgewicht / Erfolgsquote	10		0 - 100 %

Jeder Studierende kann in einem Kompetenznachweis eine Erfolgsquote von 0 bis 100% erreichen. Die gewichtete Summe aus den Erfolgsquoten pro Thema und dem Gewicht des Themas ergibt eine Gesamterfolgsquote zwischen 0 und 100%. Der gewichtete Mittelwert der Erfolgsquoten der einzelnen Kompetenznachweise wird in eine Note zwischen 3 und 6 umgerechnet. Die Note 3 (gemittelte Erfolgsquote weniger als 50%) ist ungenügend, Die Noten 4, 4.5, 5, 5.5 und 6 (gemittelte Erfolgsquote zwischen 50% und 100%) sind genügend.

10 Ergänzende Lehrmittel

Ergänzende Lehrmittel sind Empfehlungen, um sich auf das CAS vorzubereiten sowie den Stoff zu vertiefen oder zu erweitern. Weitere ergänzende Lehrmittel werden jeweils bei Start der Einzelkurse bekanntgegeben. Die Beschaffung liegt im Ermessen der Studierenden:

Nr	Titel	Autoren	Verlag	Jahr	ISBN Nr.
1.	C/C++ Das umfassende Lehrbuch	Ulrich Kaiser Martin Guddat	Rheinwerk-Verlag	2014	978-3-8362-2757-5 als E-Book
2.	Einführung in Java	Kai Günster	Rheinwerk-Verlag	2015	978-3-8362-2867-1 als E-Book
3.	Java ist auch eine Insel Einführung, Ausbildung, Praxis. Aktuell zu Java 8	Chr. Ullenboom	Rheinwerk-Verlag	2016	978-3-8362-4119-9 gratis PDF
4.	LINUX, Das umfassende Handbuch	Johannes Plötner Steffen Wendzel	Rheinwerk-Verlag	2012	978-3-8362-1822-1 gratis PDF
5.	Linux-UNIX-Programmierung	Jürgen Wolf	Rheinwerk-Verlag	2006	3-89842-749-8 gratis PDF
6.	Embedded Android	Karim Yaghmour	O'Reilly	2013	978-1-4493-0829-2
7.	Mastering Embedded Linux Programming - Second Edition	Chris Simmonds	Packt Publishing	2017	978-1787283282
8.	Linux Device Drivers Development	John Madieu	Packt Publishing	2017	978-1785280009

11 Dozierende

Vorname Name	Firma	E-Mail
Martin Aebersold	BFH	martin.aebersold@bfh.ch
Rolf Lanz	BFH	rolf.lanz@bfh.ch
Dominik Nufer	Nufer Consulting AG	dominik@nufer.org
Ivo Oesch	BFH	ivo.oesch@bfh.ch

12 Organisation

CAS-Leitung:

Ivo Oesch
Tel: +41 34 42 66 884
E-Mail: ivo.oesch@bfh.ch

CAS-Administration:

Andrea Moser
Tel: +41 31 84 83 211
E-Mail: andrea.moser@bfh.ch

Während der Durchführung des CAS können sich Anpassungen bezüglich Inhalten, Lernzielen, Dozierenden und Kompetenznachweisen ergeben. Es liegt in der Kompetenz der Dozierenden und der Studienleitung, aufgrund der aktuellen Entwicklungen in einem Fachgebiet, der konkreten Vorkenntnisse und Interessenslage der Teilnehmenden, sowie aus didaktischen und organisatorischen Gründen Anpassungen im Ablauf eines CAS vorzunehmen.

Berner Fachhochschule

Technik und Informatik

Weiterbildung

Aarbergstrasse 46

CH-2503 Biel

Telefon +41 31 848 31 11

Email: weiterbildung.ti@bfh.ch

bfh.ch/ti/weiterbildung

bfh.ch/ti/cas-ebxs