

Kevin Kettinger

Freiberuflicher Software-Entwickler

☎ +49 1764 721 50 74

✉ info@kettinger.net

🏠 Karlsruhe



Fachliche Schwerpunkte

- Embedded Software-Entwicklung in C/C++.
- Erstellung von Embedded Linux-Distributionen mit Yocto.
- Backend-Entwicklung in C++, Golang und Rust.
- Frontend-Entwicklung in Javascript, Typescript und Svelte.

Kompetenzen

- **Programmiersprachen:** C/C++, Golang, Rust, Javascript, Typescript, Python, PHP, HTML, CSS
- **Embedded:** Yocto, Linux, Kernel-Entwicklung, Device-Tree, U-Boot, Mender, RAUC
- **Frameworks:** Qt, Slint, Unified Automation, Svelte
- **Hardware-Plattformen:** PC, ARM, STM32, nRF51, ATmega, RaspberryPi, Tinkerboard, Toradex
- **Kommunikationstechnologien:** Bluetooth Low Energy (BLE), WLAN, LoRaWAN
- **Kommunikationsprotokolle:** MQTT, RabbitMQ, AMQP, OPC-UA/DA, I2C, SPI, CAN, UART, Ethernet, TCP/IP, UDP, JSON, MessagePack, HTTP
- **Datenbanken:** MongoDB, PostgreSQL, MySQL, SQLite, Oracle
- **CI/CD:** Gitlab, Jenkins, Docker, Bamboo, Bitbucket
- **Entwicklungsumgebungen:** Visual Studio Code, CLion, QT Creator, Visual Studio, Eclipse, LPCEXpresso, Keil µVision, Android Studio
- **Tools:** GIT, SVN, Docker, GDB, Valgrind
- **Sprachen:** Deutsch (Muttersprache), Englisch (C1)

Ausbildung

- **Elektrotechnik-Informationstechnik**, B. Eng., Hochschule Karlsruhe
- **Ausbildung Fachinformatiker in Systemintegration**, Göppingen

Build-Engineer

Vorwerk, Wuppertal

remote

Juni 2023 - heute

Unterstützung bei Wartung und Erweiterung einer Yocto-basierten Build-Umgebung sowie bei Entwicklung und Skalierung der zugehörigen Software-Toolchain. Durchführung von Fehleranalysen in Build-System, CI/CD-Pipeline und Endanwendung. Evaluierung und Integration von Analyse- und Coverage-Tools. Kontinuierliche Optimierung bestehender Build-Prozesse.

Sprachen C/C++, Python, Bash
Technologien Yocto, Docker, Gitlab, Jenkins, CMake, Git

Software-Developer

SWARCO, Gaggenau

remote

Februar 2021 – Juli 2024

Weiterentwicklung der Verkehrsmanagement-Software zur Steuerung und Leitung des Straßenverkehrs, Migration von klassischen Makefiles zu modernem CMake, Einführung von Software-Versionierung mit GIT. Entwicklung Tools mit Qt, Golang und C++. Unterstützung bei der Migration von Oracle zu MongoDB in der vorhandenen Softwarearchitektur, Migration der vorhandenen Daten-Kommunikation zu RabbitMQ/AMQP in der vorhandenen Softwarearchitektur, und Unterstützung bei der Containerisierung der vorhandenen Softwarearchitektur mit Docker.

Sprachen C/C++, Golang
Technologien Qt, Visual Studio, CMake, Docker, RabbitMQ, MongoDB

Software-Engineer

Faseroptik Henning, Allersberg

remote

Mai 2023

Migration der bestehenden Software des TrafficLux Touch-Eingabegerätes zur Steuerung der mobilen LED-Textanzeige für Begleitfahrzeuge von C++/Qt/QML zu Rust/Slint.

Sprachen Rust, Slint
Technologien Yocto
Plattformen Linux, Embedded, Toradex iMX6

Software-Engineer

Faseroptik Henning, Allersberg

remote / vor-Ort

November 2022 – März 2023

Konzeption und Entwicklung der Software eines Testgeräts zur Prüfung von Lichtfasern, Erstellung eines Betriebssystems auf Basis von Yocto/meta-raspberrypi, und Image-Update mit RAUC. Ansteuerung eines Schrittmotors über I2C, Entwicklung eines Frontends auf Basis von Svelte (Javascript Framework), sowie Entwicklung eines REST-Backends in Golang.

Sprachen Golang, Typescript
Technologien Svelte, Chromium, REST, Yocto, RAUC, Raspberry Pi, I2C

Software-Developer*Valeo, Friedrichsdorf***remote / vor-Ort***Juni 2022 – September 2022*

Entwicklung von Test Cases für ein bestehendes Embedded Telematik-Projekt und Erstellung der zugehörigen Dokumentation.

Sprachen	C++
Technologien	Yocto
Plattformen	Linux, Android, Embedded

Build-Engineer*MESOMATIC, Schorndorf***vor-Ort***Juni 2020 – Januar 2021*

Inbetriebnahme des STM32MP1 SoM von BytesAtWork auf einem Custom-Board, Einrichtung der Yocto Build-Umgebung, Erstellen eines Linux-Images mit Yocto, Einbindung von Peripherie in Linux/Device-Tree, darunter Ethernet PHY, LTE, WiFi und LoRaWAN. Anpassungen am Device-Tree, U-Boot und Kernel. Erstellung einer VM für Linux (A7) und Embedded (M4) Entwicklung mit der Eclipse/Yocto-Toolchain.

Technologien	Yocto, Device-Tree, Kernel, U-Boot, USB, SDIO, SPI, I2C, WLAN, LTE, LoraWAN
Plattformen	Linux, Embedded

Software-Engineer*Faseroptik Henning, Allersberg***remote***Mai 2020 – heute*

Konzeption, Neu-Entwicklung und Betreuung des TrafficLux Touch-Eingabegerätes zur Steuerung der mobilen LED-Textanzeige in Linux/Qt/C/C++ (zuvor: WinCE). Erstellung eines Linux-Images mit Yocto für die Toradex Apalis iMX6 Plattform und Integration eines LTE/GNSS Moduls über mini-PCle für die bestehende Hardware. Entwicklung einer QML-Applikation mit Funktionen für die Steuerung der LED-Anzeigentafel (RS232), Logging von Kommandos und GNSS-Positionen, Integration der Applikation in Yocto sowie Anpassung von Kernel-Parametern. Auslegung der LTE/GNSS Antennen und Realisierung von Echtzeit-Tracking über LTE/MQTT mit Visualisierung mittels der IoT-Plattform Thingsboard. Allgemeine Unterstützung in der alltäglichen Entwicklung.

Sprachen	C++
Technologien	Yocto, Device-Tree, Kernel, U-Boot, USB, SDIO, SPI, I2C, WLAN, LTE, LoraWAN
Plattformen	Linux, Embedded

Software-Engineer**remote***CST, Waldshut-Tiengen**April 2019 – Dezember 2021*

Anbindung des optischen Messsystems an eine SPS über OPC-UA, einschließlich der Entwicklung einer OPC-UA Bibliothek für weitere interne Produkte und Entwicklungen. Evaluierung und Inbetriebnahme eines OPC-DA zu OPC-UA Proxy beim Kunden und Test der Bibliothek/Anbindung vor Ort beim Kunden in der Produktion. Entwicklung eines Test-Clients/Servers zur Offline-Simulation. Entwicklung einer Rest-API (JSON) Schnittstelle für das optische Messsystem mit Authentifizierung und Rollenverwaltung. Entwicklung eines Web-Interface mit Anbindung an die Rest-API (JSON) zur Darstellung von Ergebniswerten sowie Konfiguration von Parametern. Entwicklung eines Kamera-Viewers in Qt/QML zur plattformunabhängigen Live-Darstellung von Kameradaten.

Sprachen C/C++, Python

Technologien OPC-DA, OPC-UA, Unified Automation, REST(JSON), cpprestsdk, Django, Qt/QML

Plattformen Linux, Embedded

Software-Engineer**remote***J. Bisch Engineering, Bruchsal**Januar 2019 – März 2019*

Entwicklung von E-Commerce Schnittstellen zwischen JTL-Wawi und verschiedenen Shop-Systemen und Online Buchhaltungs-Dienstleister.

Sprachen PHP

Technologien Symfony, MySQL, REST (JSON)

Plattformen Linux

Software-Engineer**remote***Domus7, Karlsruhe**Januar 2019 – Februar 2019*

Entwicklung eines Prototyps zur Aufzeichnung von Langzeitmesswerten (Temperatur, Feuchtigkeit, Luftdruck) mithilfe eines BMA280 (Bosch) Sensors.

Sprachen Python

Technologien I2C, Gnuplot, Docker

Plattformen Linux, Embedded

Software-Engineer**remote***J. Bisch Engineering, Bruchsal**April 2019 – März 2019*

Konzeption und Entwicklung einer Schnittstellensoftware für Windows-Betriebssysteme mit Qt, die Datensätze aus einer MSSQL-Datenbank an eine externe REST-API (JSON) überträgt. Design und Entwicklung eines Zugangskontrollsystems für Baustellen mit einem Web-Frontend, das über eine REST-API (JSON) an die Hardware (Dreh Sperren) angebunden ist. Dokumentation und Inbetriebnahme beim Kunden.

Sprachen C++

Technologien Qt, REST (JSON), MSSQL

Plattformen Windows

Bachelorthesis**vor-Ort***WISI Communications, Niefern-Öschelbronn**Mai 2017 – Februar 2018*

Konzeption und Entwicklung einer Android-App zur transparenten Übertragung des Web-Interfaces einer vorhandenen DVB-Kopfstele über Bluetooth LE 4. Aufbau eines Bluetooth Mesh-Netzwerkes unter Einsatz mehrerer Bluetooth-Smart On-Chip Mikrocontroller (nRF51822). Design und Entwicklung eines Netzwerkprotokolls zur Übertragung von Daten, basierend auf OpenMesh. Test und Implementierung des Protokolls im Endprodukt.

Sprachen	C/C++
Technologien	Bluetooth Low Energy, OpenMesh, mercurial, MessagePack, Qt
Plattformen	Embedded, Android

Praxissemester**vor-Ort***Rüstungsindustrie, Nahe Augsburg**März 2016 – August 2018*

Entwicklung einer GUI in Qt zur Darstellung von Radar-Daten (FMCW, Range-Doppler-Matrix) in Echtzeit aus dem Signalverarbeitungsteil, einschließlich der Implementierung eines Detektionsalgorithmus (CFAR). Test und Aufbau des Antennensystems sowie Durchführung von Antennenberechnungen.

Sprachen	C/C++
Technologien	Boost, Qt, Matlab, subversion
Plattformen	Linux, Windows

Software-Entwickler**vor-Ort***M. Rösch Engineering, Karlsruhe**September 2017 – November 2017*

Realisierung eines Tracking-Systems von Kunden anhand WLAN Probe-Requests. Implementierung der Software auf einem Raspberry-Pi.

Sprachen	C/C++
Technologien	MySQL, aircrack, WLAN, git
Plattformen	Linux, Embedded

Festanstellungen

- **Netzwerk- und Systemadministrator** bei CNS-IT in Filderstadt. Dezember 2009 - August 2012.
- **Software-Entwickler** bei Ruatti Systemtechnik in Udingen. Juli 2009 - Dezember 2009.