

Pierre Bockner

B.Sc.

Neudörferstr. 16
90402 Nürnberg
Deutschland

☎ +49 152 21868349
✉ task.pierre@posteo.de
🌐 www.bockner.com
Geboren: 09.11.1990



Ausbildung

- 10.2010–04.2015 **Bachelor**, *Technische Universität Chemnitz*, *Note: 2,0*
Informatik, Grundstudium der Informatik mit Vertiefung in Rechnerarchitektur, Compilerbau, IT-Sicherheit, Betriebssysteme und Eingebettete Systeme. Nebenfach: Physik
- 08.1997–07.2009 **Abitur**, *Roman-Herzog-Gymnasium*, Schmöln, *Note: 1,8*
Allgemeine Hochschulreife

Bachelorarbeit

- Thema *Methoden zur Objektverfolgung bei der Lokalisierung mittels Tiefenkameras*, *Note: 2,6*
- Betreuer Prof. Dr.-Ing. habil. Matthias Werner, Dipl.-Inf. Mario Haustein
- Abstrakt Tiefenkameras ermöglichen im Vergleich zur klassischen bildbasierten Lokalisierung einen zusätzlichen Parameter – die Entfernung des Objekts zur Kameraebene – zur Positionsbestimmung zu nutzen. Die Verarbeitung der Ergebnisse von Lokalisierungsverfahren können jedoch erst sinnvoll verarbeitet werden, wenn sie sich durch sog. Tracking über den Verlauf der Zeit dem lokalisierten Objekt zuordnen lassen können. Verfahrensbedingt kann es bei der Tiefenbildlokalisierung vorkommen, dass Objekte aus dem Nichts auftauchen, zu einem Metaobjekt verschmelzen, zwei eng benachbarte Objekte sich in Einzelobjekte aufteilen oder durch Überdeckung scheinbar ihre Form verändern.

Berufserfahrung

- 11.2024–heute **Softwareentwickler, Eingebettete Systeme, Zühlke Engineering GmbH**
Aufgabe Protective System für ein Klasse C IVDR Medical Device. Entwicklung eines integrierten Schutzsystems für ein IVDR Klasse C Medizinisches Gerät im Rahmen von Functional Safety Anforderungen.
Dies beinhaltet die Entwicklung eines Echtzeit Betriebssystems, um safety-kritische Monitoring- und Steueraufgaben auszuführen. Das Gerät soll den Patienten außerdem bei Fehlverhalten alarmieren.
Sprachen C
Tools CMake, Docker, DOORs, Azure DevOps
System Custom Real Time OS
Tests GoogleTest, HIL
Normen ISO 13485, ISO 14971, IEC 62304, IEC 60601
- 10.2022–heute **Softwareentwickler, Eingebettete Systeme, Zühlke Engineering GmbH**
Aufgabe Betriebssystem Entwicklung für die Steuereinheit von Solaranlagen. Entwicklung einer Zentraleinheit zur Steuerung von Solaranlagen und zur Datenerfassung von Wechselrichtern. Das Gerät hat eine Anbindung zum Netzbetreiber, um die Einspeisung des erzeugten Stroms zu kontrollieren.
Die Arbeit umfasst maßgeblich die Unterstützung des Entwicklerteams durch Bereitstellung, Migration und Pflege eines kundenspezifischen Embedded Linux Derivats als Betriebssystem (Yocto)
Sprachen C, Modern C++, Python, Shell, Ruby
Tools CMake, Docker, Ansible, Doxygen
CI/CD Gitlab
System Linux, Yocto, U-Boot, Secureboot, BSP-Integration
Tests GoogleTest, pytest
- 05.2024–07.2024 **Softwareentwickler/Tech-Lead, Eingebettete Systeme, Freiberuflich**
Aufgabe Implementation eines Kommunikationsstacks/-protokolls für RS232. Diese Aufgabe umfasst Abstimmungen und System Design mit anschließender Implementation der Stack-Layer in C.
Sprachen C
Tools CMake, GitServer, Doxygen
System CubelDE, STM-HAL
- 10.2022–06.2024 **Softwareentwickler, Eingebettete Systeme, Zühlke Engineering GmbH**
Aufgabe Entwicklung einer Firmware für einen Rauchmelder mit Licht-Sensorik Die Arbeit an der Firmware ist breit aufgestellt und fokussiert sich auf die Integration eines Bootloaders für Updates im Feld, Kommunikation mit einer Rauchmelder Zentrale (ASD), Steuer-/Messkreis für die Lichtsensorik sowie einer Persistenz für Produktions- und Laufzeitdaten.
Sprachen C, Modern C++, Python, Shell
Tools CMake, Docker, Ansible, Doxygen
CI/CD Teamcity
System LittleFs, MCUBoot, FreeRTOS, CubeMX
Tests CppUTest, pytest, HIL
Normen EN54-20

- 07.2018–09.2022 **Softwareentwickler, Eingebettete Systeme**, Zühlke Engineering GmbH
Aufgabe Anbindung landwirtschaftlicher Fahrzeuge an ein Farm Management System. Entwicklung der Firmware eines Steuergeräts für die Datensammlung auf landwirtschaftlichen Maschinen und von landwirtschaftlichen Geräten mittels CAN Bus und Isobus.
Sprachen C, Modern C++, Python, Shell
Tools CMake, Docker, MQTT, Doxygen
Interfaces Protobuf, AT-Commands (Modem), Isobus, CAN-Bus, GPS+NTRIP
CI/CD Jenkins, OSS-Licenses
System Linux, Yocto, U-Boot, Secureboot, Azure IoT Hub, Azure Blob Storage, OTA Update
Tests GoogleTest, GMock, behave
- 01.2017–07.2018 **Softwareentwickler, Eingebettete Systeme**, Diehl Metering GmbH
Aufgabe Mitentwicklung an neuen Smartmetern (Wärmezähler und Wasserzähler)
Entwicklung einer Firmware für Batterie betriebene Wasser- und Warmwasserzähler. Die Smart Meter haben eine Betriebsdauer von bis zu 12 Jahren, ohne ausgewechselt zu werden. Sie messen den Wasserverbrauch und im Speziellen den Energieverbrauch der Kunden zwischen Vor- und Rücklauf am Warmwasserzähler.
Der Zähler erfüllt die Norm EN 1434 und ist zertifiziert durch das PTB. Die Firmware wird betrieben mit einem maßgeschneiderten Betriebssystem. Ein Bootloader ermöglicht Firmware Updates im Feld über eine drahtlose IrDA Schnittstelle. Das Gerät ist sehr energieeffizient. Die Gerätekalibrierung und Prüfung auf Korrektheit der Berechnung des Wasserverbrauchs als auch des Energieverbrauchs wurde an spezialisierten Testständen durchgeführt.
Sprachen Asm, C, C++, Python, Batch
Tools svn, Eclipse (Code Composer Studio), MPLAB X, QT, Idscript, UML (Visual Paradigm), Doxygen, Latex, Jira
Controller MSP430FR6989, PIC16f1719
- 04.2016–12.2016 **Softwareentwickler, Eingebettete Systeme**, D+H Mechatronic AG
Aufgabe Mitentwicklung an der neuen Zentrale (RZN-D) für Rauch-Wärme-Abzug und Lüftungssteuerung
Für die modular aufgebaute Zentrale soll für jede Einheit eine Firmware entwickelt werden. Jedes der Module ist mit einem Bootloader ausgestattet, um ein Firmware Update aller Zentralen inklusive ihrer Module in einem Gebäude zu ermöglichen.
Die Module nutzen verschiedenste Mikrocontroller, unter anderem einen STM32F4xxx zum Betreiben einer Benutzer GUI mittels touchgfx.
Die Zentrale steuert und reagiert auf digitale Aus- und Eingänge, um z.B. Sensoren wie Rauchmelder und Wettersensoren zu betreiben oder Alarmer auszulösen. Zusätzlich werden auch Kettenantriebe zum Schließen oder Öffnen von Fenstern betrieben. Eine Zentrale wird abhängig von den Gebäudedimensionen mit mehr oder weniger Einheiten ausgestattet. Die lokale Infrastruktur ist kalibrierbar als Abbild hinterlegt. Die Zentrale ist für die Norm ISO 21927-9 zertifiziert.
Sprachen Asm, C, C++, VB, Batch
Tools git, Eclipse, Atmel Studio, Touchgfx, FreeRTOS, gcc, gdb, Idscript, make, Eagle, cppcheck, Doxygen
Controller STM32L4xxx, STM32F4xxx, AT32UC3B0512
- 04.2013–06.2013 **Wissenschaftliche Hilfskraft, Bereich Universitätskommunikation**, TU-Chemnitz
Implementierung einer Fernsteuerung (Mobil, Android, Java) für den humanoiden Roboter NAO (Server, C++); Unterstützung bei der Präsentation des Roboters.

Praktika

04.2015–07.2015 **Studienprojekt/Uni-Praktikum, Professur Technische Informatik, TU-Chemnitz**

Aufgabe Implementation einer Robotersteuerung mithilfe einer FPGA (Xilinx Spartan6)

1. Synthetisieren eines vordefinierten Mikrocontrollers auf der FPGA (VHDL)
2. Ergänzung des Mikrocontrollers um ein UART-Interface (RS-232) zur Kommunikation mit dem Roboterarm (VHDL)
3. Implementation einer Kollisionsbehandlung zur Steuerung des Arms als Software (C)

Tools C, VHDL, Xilinx ISE+ISim, Eclipse, gcc, gdb

10.2013–02.2014 **Studienprojekt/Uni-Praktikum, Professur Betriebssysteme, TU-Chemnitz**

Bezeichnung Praktikum Forschungsschwerpunkt Eingebettete selbst organisierende Systeme

Aufgabe Implementierung eines einfachen Multitasking-fähigen Betriebssystems auf einer Arm-Architektur, Verwendung des Displays und der seriellen Schnittstelle

1. Implementation einer Debugereinheit über die serielle Schnittstelle
2. Implementation eines Interrupthandlers
3. Konfiguration des Systemtimers, Auslösen von Interrupts
4. Trennung von Kernel- und User-Mode
5. Syscall Interface
6. Kooperative Prozessumschaltung
7. Signale, Mutexe, Treiber, ...

Tools C, Assembler(Armv4), gcc+ldscript, gdb

Sprachen

Deutsch Muttersprache

Englisch Fließend

Erfahrungen

Betriebssysteme	Linux, Yocto, FreeRTOS, Android, Windows
Bootloader	U-Boot, Secureboot (HAB-NXP), MCUBoot
Programmieren	Assembler (x86, STM, AVR32, MSP430, PIC), C/++, Python, Batch, Shell, C#, Java, VHDL, Angular, Typescript, Javascript
Library	LittleFS, OpenCV, Protobuf, Flatbuffers, Qt, curl, Openssl, Touchgfx (Embedded GUI)
Controller	STM32L/F4xxx, AT32UC3B0512, MSP430FR6989, PIC16f1719, ...
IDEs	VsCode, CubeIDE, Eclipse, TI - CCS, Atmel Studio, Visual Studio, MPLAB X, Xilinx ISE+LSim, ModelSim
Tools	git, svn, gcc, gdb, Idscript, make, CMake, Doxygen, Visual Paradigm (UML)
Office	AsciiDoc, Markdown, Microsoft 365, LibreOffice, L ^A T _E X, Jira, Confluence, Polarion, DOORS
Normen	Medizinische Geräte (ISO 13485, ISO 14971, IEC 62304, IEC 60601), Aspirierender Rauchmelder (EN54-20) Rauch Wärme Abzug (ISO 21927-9), Warmwasser Smart Meter (EN 1434), Isobus (11783)
Sonstiges	Zertifikat - iSAQB-CPSA-F