



KI-IoT

Holistische Open-Source-Plattform
für eingebettete Systems-on-Chip

Projekt **KI-IoT**

Schlussbericht gemäß NKBF 2017 – Teil II TIB
der Infineon Technologies AG

Zuwendungsempfänger: Infineon Technologies AG 81726 München		Förderkennzeichen: 16ME0351
Vorhabenbezeichnung: Entwicklungsplattform für vertrauenswürdige IoT-Mikrochips mit innovativem KI-Co- Prozessor – ZUSE-KI-IoT		
Teilvorhaben: Beitrag zur Holistischen Open-Source-Plattform für eingebettete Sys- tems-on-Chip mit Fokus auf KI-basierte Motorsteuerungen in Automotive		
Laufzeit des Vorhabens / Berichtszeitraum: 01.07.2021 bis 30.06.2024		
Berichtsdatum 20.12.2024		
Vorgelegt beim Projektträger: VDI/VDE Innovation + Technik GmbH Steinplatz 1 10623 Berlin		
Ansprechpartner: Lars Lindstaedt		
Vorgelegt durch: Infineon Technologies AG Dr. Stephan Eder - SFM RTF CFA Am Campeon 1-15 85579 Neubiberg		

Inhaltsverzeichnis

Inhaltsverzeichnis.....	2
1. Überblick - Ziel des KI-IoT-Vorhabens	3
Arbeitspaket 4 (AP4) „Entwicklung Testumgebung“	5
Arbeitspaket 5 (AP5) „Digitalentwurf“	5
Arbeitspaket 6 (AP6) „Firmware und Inbetriebnahmewerkzeuge“	6
Arbeitspaket 5.1 (AP5.1) „Implementierung und Integration von CPU-Kerneinheiten“ ...	8
Arbeitspaket 5.3 (AP5.3) „Open-Source-Interface-Controller (CAN und Ethernet MAC)“	9
Arbeitspaket 5.4 (AP5.4) „Lage- und Drehzahlerfassung“	10
Arbeitspaket 5.6 (AP5.6) „Entwicklung sensorloser Motorregelung“	10
Arbeitspaket 5.7 (AP5.7) „Sensorlose Regelung“	11
Arbeitspaket 6 (AP6) „Firmware und Inbetriebnahmewerkzeuge“	11
Arbeitspaket 6.2 (AP6.2) „Betriebssysteme“	12
Arbeitspaket 6.5 (AP6.5) „G-Code-Interpreter“ / „Multiachsbahnplanung“	13
Arbeitspaket 6.6 (AP6.6) „Sensorlose Motorregelung auf Basis der Software/Firmware“	14
Arbeitspaket 6.9 (AP6.9) „Testfunktionen für Qualifikation und Validierung“	17
Arbeitspaket 6.10 (AP6.10) „Anwenderwerkzeuge“	18
Arbeitspaket 6.7 (AP6.7) „Überwachung und Sicherheitsfunktionen“	19
Arbeitspaket 6.8 (AP6.8) „Predictive Maintenance“	20
2. Fazit.....	22
3. Abbildungsverzeichnis	22

1. Überblick - Ziel des KI-IoT-Vorhabens

Die Vorhabensbezeichnung des vom BMBF geförderten Forschungsprojekts lautete „Entwicklungsplattform für vertrauenswürdige IoT-Mikrochips mit innovativem KI-Co-Prozessor – ZUSE-KI-IoT“. Der Name der Teilvorhabensbezeichnung war „Holistische Open-Source-Plattform für eingebettete Systems-on-Chip mit Fokus auf KI-basierte Motorsteuerungen in Automotive“. Das Projekt-Kürzel lautete „KI-IoT“.

Ziel des KI-IoT-Vorhabens war die Entwicklung und quelloffene Bereitstellung einer modularen Plattform für den Entwurf von Systems-on-Chip (SoC) auf Basis der RISC-V-Architektur mit eingebetteten nichtflüchtigen Speichern (NVM). Der Open-Source-Hardware-Ansatz für digitale Kernkomponenten sowie Lizenzmodelle analoger und proprietärer Funktionsblöcke soll die Einstiegshürden - insbesondere in KMUs – für anwendungsspezifische Prozessoren deutlich senken. Das KI-IoT-Projekt begann am 01. Juli 2021 und endete am 30. Juni 2024. Es gliederte sich in zehn in Wechselbeziehung stehende Arbeitspakete. Während der 36-monatigen Laufzeit lagen die Schwerpunkte der Infineon-Aktivitäten auf den Arbeitspaketen 4, 5 und 6.

AP														IFAG	
		Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12		
1	Verbundprojektleitung														
4	Entwicklung Testumgebung														
4.1	Entwicklung FPGA-Testplattform	■	■												3
4.2	Entwicklung Leiterkarten zur Erprobung							■				■			3
4.3	Entwicklung Virtuelle Entwicklungsplattform	■	■	■											
5	Digitalentwurf														
5.1	Implementierung CPU-Kerneinheiten			■	■	■	■	■	■	■	■	■	■	■	6
5.2	Standardperipherie	■	■	■											3
5.3	Interface-Controller				■	■	■	■							3
5.4	Drehzahl- und Lageerfassung	■	■	■	■							■	■		6
5.5	Kompensation Nichtlinearitäten in Servoantrieben		■	■	■	■							■	■	3
5.6	Sensorlose Regelung (SFOC)				■	■	■	■	■	■	■	■	■	■	6
5.7	Bahnplanung				■	■	■	■	■	■	■	■	■	■	6
5.8	Aktive Schwingungsunterdrückung						■	■	■				■	■	
5.9	Synthese und Layout				■			■				■			
6	Firmware und Inbetriebnahmewerkzeuge														
6.1	Treiber, API und Beispielcode für Peripherie		■	■	■	■	■	■	■	■					6
6.2	Betriebssystem					■	■	■	■						3
6.3	Protokollstack für CAN und Ethernet			■	■						■	■	■	■	3
6.4	API Motorregelung				■	■	■	■	■	■	■	■	■	■	
6.5	G-Code-Interpreter Multiachsbahnplanung								■	■	■	■	■	■	12
6.6	Sensorlose Regelung auf Basis der FW		■	■	■	■	■	■							21
6.7	Überwachung und Sicherheitsfunktionen				■	■	■	■	■	■	■	■	■	■	0
6.8	Predictive Maintenance														3
6.9	Testfunktionen für Qualifikation und Validierung														3
6.10	Anwenderwerkzeuge				■	■	■	■	■	■	■	■	■	■	6

Abbildung 1: Arbeitsplan und Berichtszeiträume der Arbeitspakete (Quelle: Infineon)

Arbeitspaket 4 (AP4) „Entwicklung Testumgebung“

Der Titel von Arbeitspaket 4 (AP4) lautete „Entwicklung Testumgebung“. Unter der Bezeichnung „Entwicklung FPGA-Testplattform“ waren die AP4.1-Anforderungen zusammengefasst. Dies beinhaltete die Evaluierung von Open-Source IP-Blöcken sowie den Entwurf von Entwicklungsboards auf Basis von Field Programmable Gate Arrays (FPGAs). An das FPGA schloss das Infineon-Projektteam die Peripherie zur Motorregelung und Lageerfassung an und es verband die Schnittstellencontroller mit dem FPGA. Das Prozessorsystem konfigurierte das Team über ein externes JTAG-Interface. Für USB, CAN und Ethernet griffen die Projektmitarbeiter auf entsprechende Transceiver zurück. Zur Inbetriebnahme stimmten sie Hardwareentwurf, Digitaldesign und Softwareentwicklung eng aufeinander ab.

Das Infineon Projekt-Team wählte das Versal-AI-Edge-Board von Xilinx als FPGA-Plattform aus, auf der das Vitis AI Software-Framework zum Einsatz kam. Das Evaluierungsboard umfasste die FPGA-Architektur zusammen mit I/Os wie On-Chip-DRAM und Flash. Konnektivität wurde über Ethernet, USB oder Breakout-Ports realisiert, die über Mezzanine-Anschlüsse zugänglich waren. Das Board verfügte über PCIe-Konnektivität, war alternativ aber auch als Lösung mit Fast Ethernet oder USB zu betreiben. Die Auswahl des AI-Edge-Boards kam deshalb in Betracht, weil es ein ausgewogenes Verhältnis zwischen Leistung und Strombedarf, geringer Latenz, Größe und thermischen Beschränkungen sowie Sicherheit und Zuverlässigkeit bot. Anforderungen in diesem Zusammenhang waren die heterogene Verarbeitung, gehärtete Infrastruktur, funktionale Sicherheit sowie Flexibilität und Programmierbarkeit.

Das Board bot die geforderte hohe Leistung beim Ablauf der Anwendung - vom Sensor über KI bis hin zur Echtzeitsteuerung. Die Architektur der AI-Engine basierte auf einem skalierbaren Array aus Vektorprozessoren und verteiltem Speicher, wohingegen die DSP-Engine über eine Slice-Architektur und einen adaptiven SoC mit integrierter Fließkomma-Unterstützung verfügte, die sich für Wireless- und Bildsignalverarbeitung, Datenanalyse und Bewegungssteuerung eignete. Die Architektur erfüllte zudem Sicherheitsanforderungen und kritische Standards wie ISO 26262 und IEC 61508.

Arbeitspaket 5 (AP5) „Digitalentwurf“

Das Arbeitspaket 5 (AP5) „Digitalentwurf“ bestand aus neun Teilarbeitspaketen. Arbeitspaket 5.2 adressierte die Standardperipherie. Microcontroller benötigen Standard-Peripherieblöcke zur Kommunikation und Datenerfassung. Sie bilden das Kernstück eines Funktionsmoduls, das externe Peripherieeinheiten bedient. Zur Kommunikation zwischen Peripherieeinheiten setzte das Infineon-Projektteam standardisierte Interfaces ein. Hierzu gehörten serielle Interfaces wie z.B. UART (Universal Asynchronous Receiver Transmitter) zur Kommunikation mit anderen Microcontrollern. SPI (Serial Peripheral Interface) und I2C (Inter-Integrated Circuit) nutzten die Teammitarbeiter zur Kommunikation mit externen Bauelementen wie z.B. ADCs (Analog-Digital Converter), DACs (Digital-Analog Converter), externen Speichern oder integrierten Sensoren.

Im AP5.4 analysierte das Infineon-Projektteam Lösungen, deren Einsatz die Positions- und Drehzahlerfassung von E-Motoren optimiert. Industrietypische Feedbacksysteme wurden für die Servoregelung durch digitale Funktionsblöcke ausgewertet. Neben klassischen Inkrementalgebern sind insbesondere digitale Hallsignale bei Brushless DC-Motoren (BLDC) bzw. EC-Motoren weit verbreitet. Durch den Einsatz von modellbasierten Beobachterstrukturen ließ sich die Positions- und Drehzahlerfassung deutlich verbessern. Zusätzlich wurden nichtlineare und systematische Abbildungsfehler der Positionsgeber korrigiert. Abbildungsfehler entstehen beispielsweise durch Montagetoleranzen, Sensorkennlinien und Verzögerungen bei digitaler Signalverarbeitung. Das Arbeitspaket sah vor, beobachterbasierte Drehzahl-Erfassungsmethoden für den Einsatz in den Demonstratoren zu erproben und zu bewerten. Bewertungskriterien für das Team waren der Aufwand bei der Inbetriebnahme, die Rechenzeitanforderungen bzw. der Logikaufwand sowie die erreichbare Dynamik bei gegebener Genauigkeit.

Im Arbeitspaket 5.5 fokussierten sich die Team-Mitarbeiter auf das Thema „Kompensation von Nichtlinearitäten in Servoantrieben“. Bestimmte Komponenten eines Servoantriebs weisen fertigungs- und technologiebedingte Nichtlinearitäten auf. Beispielsweise ist die Spannungsabbildung durch eine schaltende Endstufe auf MOSFET-Basis nur begrenzt linear. Eine Stromregelung kann diese Effekte nur begrenzt justieren, da gerade in dynamischen Lastfällen bzw. beim Stromnulldurchgang immer wieder Stromabbildungsfehler auftreten, die zum Drehmoment-Ripple und unruhigem Motorlauf führen. Ursächlich für diese Effekte sind bestimmte Eigenschaften der Halbleiterelemente, die sich über Lebenszeit, Temperatur und Laststrom verändern. Zur Modellierung dieser Effekte entwarf das Team eine Simulation der schaltenden Endstufe zur Abbildung dieser Effekte. Zur Kompensation der Nichtlinearität entwarf das Projektteam selbststellende Kompensationsverfahren und implementierte sie in Hardware. Die Verfahren wurden an mehreren Aufbauten validiert. Das Ziel war die Implementierung ressourcenoptimaler Kompensationsverfahren, die sich mit minimalem Hardwareaufwand umsetzen ließen. Ferner ging es den Mitarbeitern im AP5.5 um die Evaluierung von Möglichkeiten, einen rotorpositions-abhängigen Drehmoment-Ripple marktgängiger BLDC-Motoren durch Tabellenkompensation zu reduzieren.

Arbeitspaket 6 (AP6) „Firmware und Inbetriebnahmewerkzeuge“

Für AP6 lautete das Ziel, Firmware und Inbetriebnahmewerkzeuge zu evaluieren. Im AP6.1 war die Forderung an das Infineon-Projektteam, Treiber und Application Programming Interfaces (APIs) für Peripherie-Einheiten der Prozessorplattform zu entwickeln. Die Arbeiten umfassten ebenso die Bereitstellung von Software für Open-Source-Peripherieblöcke. Die Softwareentwicklung der Treiber erfolgte parallel zur Implementierung der Digitalblöcke. Dies ermöglichte direktes Feedback an die Digitalentwickler zur Schnittstellen- und Funktionsoptimierung der Blöcke. Diese Arbeitsschritte waren geplant, bevor die Funktionsblöcke in den Fertigungsdurchläufen der Wafer als dedizierte Hardware realisiert und fixiert wurden.

Den Infineon-Schwerpunkt im AP6.6 bildete die Entwicklung und Software-Implementierung einer sensorlosen Motorregelung auf RISC-V-Basis und die Implementierung in Neural Networks (MLP, LSTM etc.). Einerseits diente dies dem Projektteam als Performance-Analyse und Benchmark gegenüber der in AP5.6 entwickelten Hardware-Implementierung.

Andererseits erlaubte die Softwarerealisierung eine einfachere und schnellere Ersterprobung und Prüfung von Algorithmen zur sensorlosen Kommutierung von Motoren. Ein weiterer Aspekt der Softwarerealisierung war die Flexibilität in der Gestaltung und Auslegung eines Systems innerhalb des Baukastens für eingebettete Microcontroller. Sie erlaubte den Entwicklern die Realisierung von anwendungsspezifischen Systemen, bei denen nicht höchste Leistungsfähigkeit der Hardware im Vordergrund stand, sondern stattdessen auf möglichst kleine Siliziumflächen und niedrige Kosten Wert gelegt wurde.

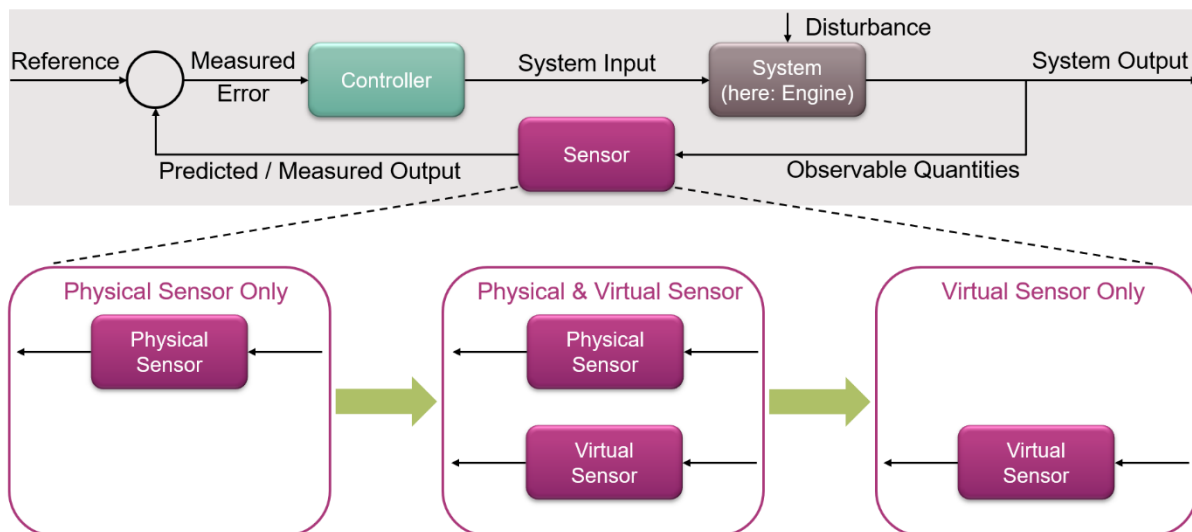


Abbildung 2: Sensorlose Motorregelung (Quelle: Infineon)

Ein Ziel des KI-IoT-Vorhabens war die Entwicklung und Bereitstellung einer modularen Plattform für den Entwurf von Systems-on-Chip (SoC) auf Basis der RISC-V-Architektur mit eingebetteten nichtflüchtigen Speichern (NVM). Die SoCs wurden insbesondere für Edge-Anwendungen in IoT-Netzwerken konzipiert. Während derartige Geräte oftmals auf Sensoranwendungen fokussieren, setzte das Infineon-Team in diesem Projekt verstärkt auf Aktorik. Dazu entwickelten die Team-Mitarbeiter Motorregelungs-Funktionen und analoge Peripherie, die zur hohen Integration von Plattformprodukten beitragen. Kernpunkte der Motorregelungs-Funktionsentwicklung war die Steigerung von Dynamik, Genauigkeit sowie Effizienz der Antriebe. Daneben evaluierte das Team Möglichkeiten zur Kostensenkung durch Sensorreduktion. In diesem Zusammenhang kamen auch KI-basierte Ansätze zur Virtualisierung von Resolvern in E-Antrieben zur Anwendung.

Ein weiteres Projektziel war der Einsatz von KI-Methoden zur Regelung. Neuronale Netze, Support Vector Machines oder Gauß'sche Prozesse boten hierfür gute Ansätze. Die simulative Validierung einer verbesserten Motorregelung mit KI-Methoden schaffte eine gute Voraussetzung für weitere Versuche und Analysen. Die neue Generation von Infineon's Mikrocontroller-Serie AURIX bot mit den integrierten Funktionsblöcken und performanten Verarbeitungsmerkmalen (Hardwarebeschleuniger) hierfür die ideale Basis. Die Hardwarebeschleuniger unterstützten hierbei kosten- und zeiteffiziente Implementierungen.

Der Schwerpunkt der Arbeiten des KI-IoT-Projekts lag für Infineon im Berichtszeitraum des Jahres 2022 auf den Arbeitspaketen 5 und 6. Im Rahmen dieses Projektabschnitts fokussierte sich das Infineon-Team darauf, E-Motorsteuerungen sowohl für Automotive-Anwendungen

als auch für Antriebe in der Industrieautomatisierung mit neuen IC-Lösungen zu bestücken. Dafür wurden Silizium-basierte Halbleitertechnologien in puncto Systemintegration und Zuverlässigkeit von KI-Methoden erforscht. Infineon arbeitete eng mit Partnern zusammen, die zur Optimierung des Gesamtsystems beitrugen und durch Synergien die Entwicklung neuartiger Lösungen unterstützten.

Arbeitspaket 5.1 (AP5.1) „Implementierung und Integration von CPU-Kerneinheiten“

Arbeitspaket 5 (AP5) des KI-IoT-Projekts bestand aus mehreren Teilarbeitspaketen. AP5.1 adressierte die „Implementierung und Integration von CPU-Kerneinheiten“. Anhand der in AP2 durchgeführten Anforderungsanalyse skizzierte das Projektteam eine Architektur und plante ihre Umsetzung in konkrete Hardwareeinheiten.

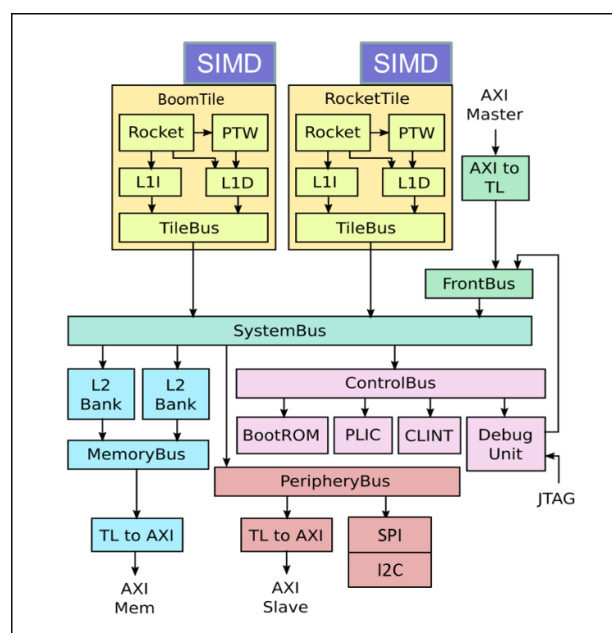


Abbildung 3: SIMD-Befehlssatzerweiterung des RISC-V-Cores (Quelle: UzL/Infineon)

Die Entwickler definierten CPU-interne Funktionseinheiten, zu denen Registerfile-Größen und interne Bus-Größen, Cache-Organization, Cache-Hierarchie und Cash-Controller, MMU (Speicher und I/O), Interruptverhalten und Pipeline-Organisation, Fehlertoleranz-Mechanismen und sicherheitsspezifische Mechanismen sowie dedizierte Funktionseinheiten wie z.B. FPU, SIMD und DSP/MAC nebst ISA-Erweiterung zählten.

Die Aufgaben im AP5.1 betrafen vor allem die Verbindung der On- und Off-Chip-Komponenten. Das Team plante, die On-chip-Peripherie über einen breitbandigen Bus an die CPU anzubinden. Hierfür gab es zwei Herangehensweisen. Das PULP-Framework setzte auf den AMBA-Bus, welcher aufgrund seiner Verbreitung auch im KI-IoT-Projekt Anwendung fand. Das CHIPYARD-Framework wiederum setzte auf den TileLink-Standard. Ein dediziertes Interfacing zu DRAM und Flash wurde für die Off-Chip-Anbindungen geplant. Soweit erforderlich, adressierte dieses AP nicht nur das Interfacing zu AMBA, sondern beinhaltete auch ein eventuelles Bridging zu anderen Bussystemen. Die in den jeweiligen Frameworks

enthaltenen Interface-Komponenten wurden an die Projektanforderungen angepasst. Gemäß Projektplanung beinhaltete dies die Bereitstellung einer Anbindung an den AXI-Bus.

Die Entwickler verwendeten Timereinheiten als Zeitgeber und Erzeuger von Pulsmustern. Sie nutzten sie beispielsweise zur Pulsweitenmodulation (PWM), zur Encoderauswertung und für zyklische Aufrufe einer Funktion. Auch hier war es maßgeblich entscheidend, wie die Blöcke ausgestaltet waren, um sie in einer Vielzahl von Anwendungen zu nutzen. Für Motorregelungs-Anwendungen war insbesondere die PWM-Erzeugung vorrangig, die über eine Schalter-Totzeitvorgabe und synchronen Betrieb mehrerer PWM-Blöcke verfügt, um eine Endstufe direkt anzusteuern. Die Entwickler achteten darauf, dass der Interrupt-Controller die Anforderungen (Requests, IRQs) der Peripherieeinheiten entsprechend den Vorgaben verarbeitete. Der Interrupt-Controller priorisierte die IRQs und veranlasste die CPU zur Verarbeitung der Interrupt-Service-Routine. Die CPU löschte abschließend den Interruptstatus und kehrte zur Hauptroutine zurück. Eine Priorisierung und schnelle Verarbeitung von IRQs, sowie eine Regelung im Hinblick auf geschachtelte Interrupts waren entscheidend für die Echtzeitfähigkeiten des Mikrocontrollers.

Arbeitspaket 5.3 (AP5.3) „Open-Source-Interface-Controller (CAN und Ethernet MAC)“

Bei der Bearbeitung der Anforderungen für AP5.3 „Open-Source-Interface-Controller (CAN und Ethernet MAC)“ betrachteten die Entwickler die CAN-Schnittstelle (CAN: Controller Area Network). CAN ist ein weltweit anerkannter Feldbus-Standard zur industriellen Kommunikation. Mit CAN-FD (FD: Flexible Data-Rate) sind Datenraten oberhalb von 1Mbit/s möglich. Ursprünglich für den Einsatz im Automotive-Umfeld entwickelt, haben Kosten, Performance und Verfügbarkeit von Komponenten den CAN-FD-Bus für eine Vielzahl von Anwendungen attraktiv gemacht. Da es Open-Source-Implementierungen von CAN-Controllern gibt, die als Peripherieeinheiten in Microcontroller integriert werden können, entschied sich das Infineon-Team für diese Variante. Das Team plante, in der KI-IoT-Implementierung diese Controller zu nutzen, da sie das Versenden und Empfangen von Daten eigenständig übernehmen. Der Microcontroller stellt beispielsweise Daten in einem Datenpuffer bereit und der CAN-Controller versendet diese automatisch nach Auslösen der Kommunikation durch die CPU. Empfangene Daten werden ebenfalls in einem Puffer für den Microcontroller bereitgestellt. So ist es möglich, den Kommunikationsablauf bis zu einer relativ hohen Ebene vollständig durch den CAN-Controller abzubilden. Ein weiteres Ziel des Arbeitspakets war es, einen vorhandenen OpenSource-CAN-Controller in die Plattform zu integrieren. Dabei war den Projektmitarbeitern die Skalierbarkeit wichtig, da die Blöcke stark konfigurierbar sind.

Um eine kostengünstige Lösung zu realisieren, wurde evaluiert, ob die Datenpuffer des Controllers in einem separaten SRAM-Speicherblock oder in Form von Flipflops direkt im Digitalteil implementiert werden konnten. In der industriellen Kommunikation werden zunehmend Ethernet-basierte Kommunikationsbusse eingesetzt. Diese beruhen auf der Bitübertragungsschicht (Physical Layer) von Ethernet, implementieren aber angepasste Protokolle auf der Sicherungsschicht (MAC Layer). EtherCAT ist beispielsweise eine Technologie, die ein spezielles Echtzeitprotokoll über einen physikalischen Ethernetbus überträgt. Die EtherCAT-Technologie gewährleistet sehr hohe Echtzeitanforderungen,

verursacht aber gleichzeitig einen hohen Aufwand durch Spezialhardware. Die Infineon-Entwickler fanden heraus, dass für kostengünstige IoT-Anwendungen mit niedrigeren Echtzeitanforderungen die Integration einer einfacheren Ethernetschnittstelle sinnvoller ist. Ein Ethernet-MAC-Controller wird hierzu in die Hardware integriert. Mit 100 Mbit/s bietet die Schnittstelle eine hohe Bandbreite.

Arbeitspaket 5.4 (AP5.4) „Lage- und Drehzahlerfassung“

Im Arbeitspaket 5.4 „Lage- und Drehzahlerfassung“ befassten sich die Infineon-Entwickler mit industrietypischen Feedbacksystemen, die für die Servoregelung durch digitale Funktionsblöcke ausgewertet werden. Neben klassischen Inkrementalgebern sind digitale Hallsignale bei permanenterregten Synchronmotoren (PMS: Permanent Magnet Synchronous Machines) weit verbreitet. Durch den Einsatz von modellbasierten Beobachterstrukturen lässt sich die Positions- und insbesondere die Drehzahlerfassung deutlich verbessern. Zusätzlich können nichtlineare und systematische Abbildungsfehler der Positionsgeber korrigiert werden. Im Rahmen des KI-IoT-Projekts erprobten die Entwickler beobachterbasierte Drehzahlerfassungsmethoden für den Einsatz in den Demonstratoren. Bewertungskriterien waren der Inbetriebnahme-Aufwand, die Rechenzeitanforderungen bzw. der Logikaufwand sowie die erreichbare Dynamik bei gegebener Genauigkeit.

Arbeitspaket 5.5 hatte die „Kompensation von Nichtlinearitäten in Servoantrieben“ zum Thema. Komponenten des Servoantriebs wiesen fertigungs- und technologiebedingte Nichtlinearitäten auf. Beispielsweise war die Spannungsabbildung durch eine schaltende Endstufe auf MOSFET-Basis nur begrenzt linear. Die Stromregelung konnte diesen Effekt nur begrenzt ausregeln, da gerade in dynamischen Lastfällen bzw. beim Nulldurchgang immer wieder Abbildungsfehler auftraten, die zu Drehmomentripplern und unruhigem Motorlauf führten. Ursächlich für diese Effekte waren Eigenschaften der Halbleiterelemente, die sich über die Lebenszeit und Temperatur sowie über den Laststrom veränderten. Zur Modellierung dieser Effekte entwarf und validierte das Infineon-Team eine Simulation der schaltenden Endstufe, womit sich dieser Effekt abbilden ließ. Zur Kompensation der Nichtlinearität entwarfen sie selbsteinstellende Kompensationsverfahren und implementierten sie in Hardware. Die Verfahren wurden an mehreren Aufbauten erprobt. Das Ziel war die Implementierung ressourcenoptimaler Kompensationsverfahren, die mit minimalem Hardwareaufwand umsetzbar sind und Parametriermöglichkeiten bieten.

Arbeitspaket 5.6 (AP5.6) „Entwicklung sensorloser Motorregelung“

Die Arbeiten im AP5.6 ließen sich unter dem Thema „Entwicklung sensorloser Motorregelung“ zusammenfassen. Für die klassische Servoregelung eines elektrischen Antriebs sind mehrere digitale Funktionsblöcke erforderlich. Hierzu gehören u.a. ADC-Auswertung, Pulsweitenmodulation (PWM), Motorregelung, Bahnplanung und Ablaufsteuerung. Ein Ziel des Forschungsvorhabens war die Umsetzung von Funktionen zur Bewegungsführung und -planung (Motion Control) und zur Motorregelung (Motor Control). Jede Motorregelung für einen elektrischen Antrieb nutzt eine Stromregelung. Für mehrphasige Motoren wird diese oftmals als feldorientierte Regelung ausgeführt. Diese nutzt neben den Strangströmen der Maschine auch die Rotorlage, die durch einen Lagegeber

erfasst wird. Dieser Lagegeber verursacht hohe Kosten und stellt eine kritische Systemkomponente dar, wenn das Verfahren auf die korrekte Funktion des Gebers angewiesen ist.

Die Nutzung sensorloser Regelverfahren für permanenterregte Zwei- und Dreiphasen-Motoren ist erfolgsentscheidend für eine Vielzahl von Anwendungen. Daneben können die Verfahren zur Aufwertung von günstigen Gebersystemen mit niedriger Auflösung genutzt werden. Sie erlauben beispielsweise einen Betrieb bei sehr hohen Drehzahlen, in denen das Lagegebersystem versagt. Diese Regelverfahren sind nicht auf einen Drehzahl- oder Lagesensor für eine hinreichend gute Drehzahlregelung angewiesen. Zum Erschließen dieser Verfahren gehört der Aufbau von PWM-Blöcken, die eine sensorlose Regelung unterstützen. Diese Korrekturregelung greift ein, wenn Modell- und Messgrößen voneinander abweichen. Die Regelungsstruktur wird in diesem Fall gleichzeitig zur Identifikation der Motorparameter genutzt. Motorparameter sind in diesem Fall Wicklungsinduktivität, Wicklungswiderstand und Permanenterregung neben den mechanischen Streckenparametern Massenträgheit und Reibungskonstante. Die Qualität der Parameteridentifikation ist dabei entscheidend für die Regelqualität in Bezug auf Dynamik und Effizienz. Zur Validierung wird die sensorlose Drehzahlregelung mit einer Servoregelung mit Lagegeber verglichen. Ein Vorhabensziel war die Implementierung selbsteinstellender Verfahren, die beim späteren Anwender minimales Motorregelungswissen erfordern. Dieser Vorteil wurde durch die Entwicklung intuitiv bedienbarer Inbetriebnahmesoftware (AP6.10) unterstützt.

Arbeitspaket 5.7 (AP5.7) „Sensorlose Regelung“

Das im AP5.7 ursprünglich definierte Aufgabenpaket zum Thema „Bahnplanung“ wurde gemäß der TVB-Version vom 26.03.2021 ersetzt durch die im AP6 von Infineon beschriebenen Aktivitäten zum Thema „Sensorlose Regelung“. Der Demonstrator stammte aus dem automobilen Anwendungsbereich und adressierte Servo-Antriebe für Nebenaggregate. Dieses Einsatzfeld stellte besondere Anforderungen an funktionale Sicherheit, Robustheit, Echtzeitfähigkeit und Lebensdauer, aber auch an Energieeffizienz und Kosten. Diese Themen wurden vom Projektteam im AP6 näher betrachtet und beschrieben.

Arbeitspaket 6 (AP6) „Firmware und Inbetriebnahmewerkzeuge“

Das Arbeitspaket 6 (AP6) „Firmware und Inbetriebnahmewerkzeuge“ bestand aus zehn Teilarbeitspaketen (AP6.1 bis AP6.10), die teilweise vom Infineon-Projektteam adressiert wurden. Das Arbeitspaket hatte zum Ziel, Firmware und Werkzeuge für Demonstratoren und Plattformen zu untersuchen und zum Projektende bereitzustellen. Im Arbeitspaket 6.1 wurden Treiber und API (Application Programming Interface) für Peripherieeinheiten der Prozessor-Plattform entwickelt. Das Infineon-Team evaluierte Software für Open-Source-Peripherieblöcke und stellte sie als Open Source für die Plattform bereit. Das im AP5.3 gewonnene Know-how diente dazu, Standard-Peripherieblöcke zu entwerfen. Hierzu zählten UART, SPI, I2C, Timer und Interrupt-Controller. Die Softwareentwicklung der Treiber wurde parallel zur Implementierung der Digitalblöcke durchgeführt. Dies ermöglichte direktes Feedback an die Digitalentwickler zur Schnittstellen- und Funktionsoptimierung der Blöcke während der Entwicklung auf den FPGA-Testboards.

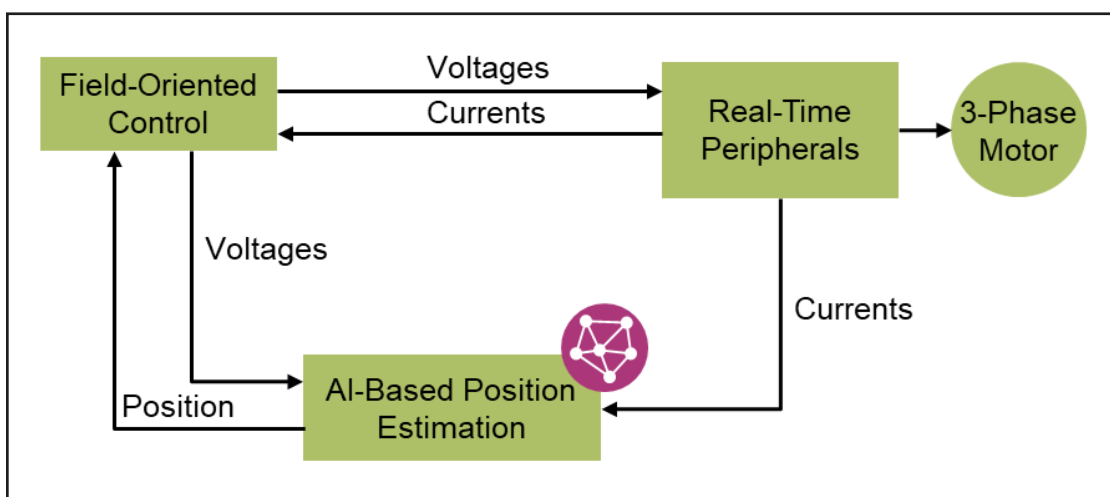
Im nächsten Schritt entwickelte das Infineon-Team Beispielcode für diese Blöcke. Diese stellten Nutzungsbeispiele der jeweiligen Blöcke dar und dienten den späteren Anwendern als Referenz und Ausgangsbasis für eigene Implementierungen. Die Software wurde entsprechend dem MISRA-C-Programmierstandard geschrieben (MISRA: Motor Industry Software Reliability Association), um die hohen Qualitätsanforderungen der Automobilindustrie zu erfüllen. Ein Entwickler-Team bestehend aus mehreren Software-Ingenieuren stellte mit regelmäßigen Prüfungen sicher, dass die in C-Code programmierten Softwareabläufe für Dritte nachvollziehbar blieben.

Arbeitspaket 6.2 (AP6.2) „Betriebssysteme“

Gemäß den Vorgaben für AP6.2 untersuchte das Projektteam verschiedene Betriebssysteme mit Blick auf die geplanten Demonstratoren. Das infrage kommende Betriebssystem sollte auf einer quelloffenen, lizenzfrei verfügbaren Version basieren, beispielsweise RTOS, das die Ausführung von Protokollstacks wie TCP/IP und UDP ermöglichte. Zur Auswahl standen unter anderem FreeRTOS oder a:FreeRTOS sowie Embecosm oder (RT-)Linux. Gleichzeitig sollte die BS-Version das Scheduling von verschiedenen Background-Tasks ermöglichen, die beispielsweise KI-Algorithmen für Predictive Maintenance ausführen. Als weitere BS-Anforderungen führte das Entwickler-Team minimale Speicher- und Rechenzeitanforderungen auf. Nach Auswahl der passenden Lösung begannen die Entwickler, das Betriebssystem auf die Testhardware zu portieren.

Im Teilarbeitspaket 6.3 begann das Infineon-Team damit, Protokolle für die Schnittstellen CAN und Ethernet auf Basis existierender Open-Source-Implementierungen umzusetzen. Als Hardwarebasis dienten die im AP5 „Digitalentwurf“ erprobten Open-Source-Peripherieblöcke. Diese Implementierung diente auch als Basis für weitere Protokolle. Für Ethernet wurden die IP-Protokolle UDP (User Datagram Protocol) und TCP (Transmission Control Protocol) implementiert, die auf dem MAC-Layer des Ethernet-Controllers aufsetzen. Als Beispielanwendung implementierte das Projektteam einen Webserver, der für Firmware-Updates oder für Statusanzeigen eingesetzt und als Ausgangsbasis für IoT-Anwendungen genutzt wurde.

Zur Inbetriebnahme und späteren Nutzung der MCU-Plattform stellten die Entwickler gemäß den Anforderungen in AP6.4 „API für Motorregelung“ für jeden Hardwareblock



Softwareunterstützung in Form von Programmierschnittstellen (APIs: Application Programming Interfaces) bereit, auf welche die jeweiligen Anwendungen zugriffen. Die API bildeten mit dem entsprechenden Hardwareblock eine Einheit und reduzierten den Aufwand bei der Implementierung und Nutzung einer Anwendung. Sie abstrahierten die zugrundeliegende Hardware für den Applikationsentwickler.

Abbildung 4: AI-basierte sensorlose feldorientierte Motorregelung (FOC) (Quelle: Infineon)

Der KI-basierte Positionssensor trägt zur Verbesserung der Verfügbarkeit, Kostenreduktion und Genauigkeit bei. Zu den Versuchen der Entwickler zählte die Ausführung eines neuronalen Netzwerks, dessen Regelzyklus eine Frequenz von 20 kHz hatte. Die Ausführungszeit des neuronalen Netzwerks mit 150 Neuronen benötigte auf einer 400 MHz basierten skalaren RISC-CPU ca. 100µs. Zur Ausführung setzte das Team einen KI-Beschleuniger mit einer Kapazität von Faktor 10 ein.

Die rot umrandete Box in der nachfolgenden Abbildung enthält Software-Funktionen, mit denen API abstrahiert wurden. Schnittstellen und Übergabeparameter sind in der API abgebildet.

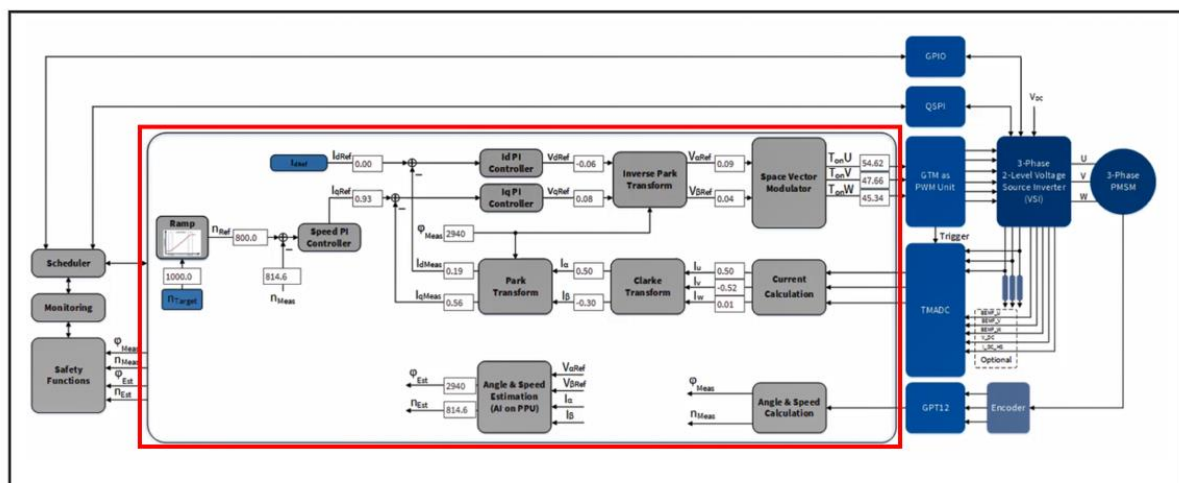


Abbildung 5: Schaltdiagramm einer Extended MLP Motorregelung (Quelle: Infineon)

Arbeitspaket 6.5 (AP6.5) „G-Code-Interpreter“ / „Multiachsbahnplanung“

Arbeitspaket 6.5 hatte die Themen „G-Code-Interpreter“ und „Multiachsbahnplanung“ zum Ziel. Unter Nutzung der Bahnplanungsfunktionen für einzelne Achsen planten die Infineon-Entwickler, dies auf mehrere miteinander synchronisierte Achsen anzuwenden, wie sie zum Beispiel in der Robotik oder beim Einsatz in der CNC-Fertigung zum Einsatz kommen. Dazu war die Entwicklung und Implementierung mehrdimensionaler Algorithmik notwendig, die von der jeweiligen Kinematik des Mehrachssystems abhing. Das Infineon-Team wählte einen als Open Source verfügbaren G-Code-Interpreter aus und passte ihn bedarfsgerecht an. Die

Verarbeitungspuffer des G-Code-Interpreters wurden von einem Slicer mit G-Code-Kommandos gefüllt, die der Slicer z.B. aus einem 3D-CAD-Modell eines herzustellenden Objekts erzeugte. G-Code ist eine definierte und standardisierte Repräsentation von Maschinen- und Bewegungsbefehlen. Ein 3D-CAD-Modell oder andere mehrdimensionale Daten werden dabei auf einzelnen Teilstrecken der verschiedenen Dimensionen heruntergebrochen. Der Interpreter übersetzt sie auf die zugrundeliegende Hardware. Die Funktionalität wurde vom Entwickler-Team in einer mehrachsigen Anwendung demonstriert.

Arbeitspaket 6.6 (AP6.6) „Sensorlose Motorregelung auf Basis der Software/Firmware“

Der Schwerpunkt im Arbeitspaket 6.6 „Sensorlose Motorregelung auf Basis der Software/Firmware“ war die Entwicklung und Implementierung einer sensorlosen Motorregelung auf RISC-V-Basis bzw. als Implementierung in MLP-Netzen (MLP: Multilayer Perzeptron). Das Entwickler-Team nutzte die Ergebnisse als Performance-Analyse und Benchmark gegenüber der im AP5.6 entwickelten Hardwareimplementierung. Darüber hinaus erlaubte die Softwarerealisierung ihm eine schnelle Erprobung und Prüfung der Algorithmen zur sensorlosen Kommutierung von Motoren.

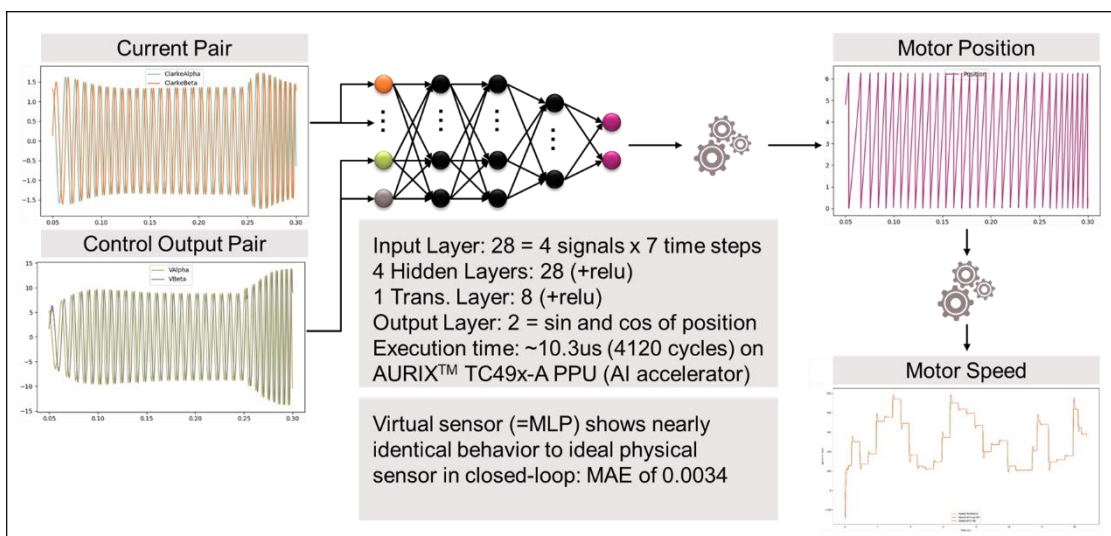


Abbildung 6: AI-basierter Positionssensor für Field Oriented Control (FOC) (Quelle: Infineon)

Der KI-basierte Positionssensor trug zur Verbesserung der Verfügbarkeit, Kostenreduktion und Genauigkeit bei. Zu den Versuchen des Projektteams zählte die Ausführung eines neuronalen Netzwerks, dessen Regelzyklus eine Frequenz von 20 kHz hatte. Die Ausführungszeit des neuronalen Netzwerks mit 150 Neuronen benötigte auf einer 400 MHz basierten skalaren RISC-CPU ca. $100 \mu s$. Zur Ausführung wurde ein KI-Beschleuniger mit einer Kapazität von Faktor 10 eingesetzt. Der virtuelle Sensor (Multilayer Perceptron: MLP) verhielt sich im geschlossenen Regelkreis nahezu identisch zum idealen physischen Sensor. Der mittlere absolute Fehler (MAE: Mean Absolute Error) betrug 0,0034. Die Komplexität des

MLP-basierten virtuellen Sensors wurde in Multiplikations- und Akkumulationsoperationen (MAC) betrachtet. Die MLP-Schicht hatte eine Komplexität von 784 MAC. Insgesamt betrug die MLP-Komplexität 3976 MAC. Der Vergleich zeigt, dass beim MLP-Einsatz höchste Genauigkeit bei kürzester Laufzeit erzielbar ist.

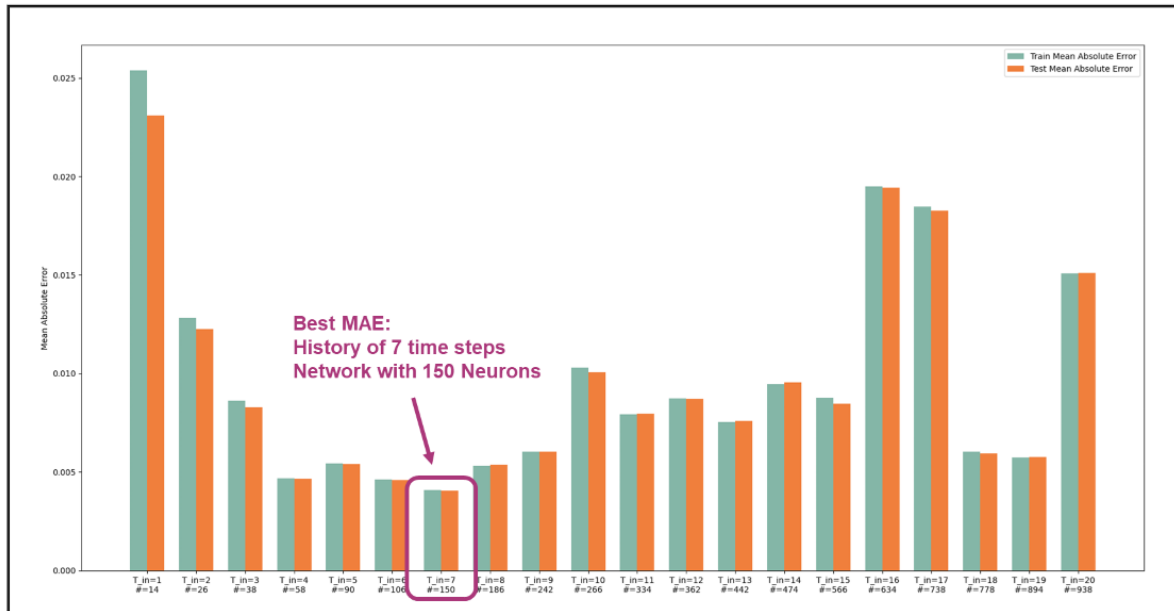


Abbildung 7: Mittlerer absoluter Fehler (MAE) über Eingabehistorie (Quelle: Infineon)

Die Universität zu Lübeck (UzL) als Infineon-Partner im KI-IoT-Projekt untersuchte eine Vielzahl von IPs sowohl aus dem kommerziellen Bereich (Xilinx), als auch aus dem Open-Source-Umfeld. Die UzL-Entwickler bevorzugten Open-Source-IPs. Diese Entscheidung beeinflusste die Auswahl des Chipyard-Frameworks für die Entwicklung und SoC-Generierung einschließlich der Kerne, Busse, Cache-Speicher und der externen Konnektivität über AXI und TL. Der von UzL-Entwicklern durchgeführte RISC-V-Core-Benchmark führte zur Auswahl des Rocket-Chip-Cores. Es wurden zwei RISC-V Rocket-Chips ausgewählt: ein 64-Bit-Core mit MMU, um ein potenzielles Mehrzweck-Betriebssystem (Linux) auszuführen, und ein 32-Bit-Core ohne MMU, um das Verhalten einer Echtzeit-Mikrocontroller-Einheit zu erzielen.

Die Allgegenwärtigkeit des Linux-Betriebssystems ermöglichte eine Abstraktionsschicht für die Hardware über generische Geräteaufwerke, mit denen die meisten Hardwarekomponenten direkt aktiviert werden konnten. Auf UzL-Ebene wurden einige Hardware-/Softwarekomponenten angepasst: Die Entwickler erstellten ein sicheres und platinen-übergreifendes Kommunikationsprotokoll über i2c. Außerdem konzipierten sie eine eng gekoppelte SIMD-Einheit, die mit den Anforderungen der RISC-V-Vektorerweiterung kompatibel war.

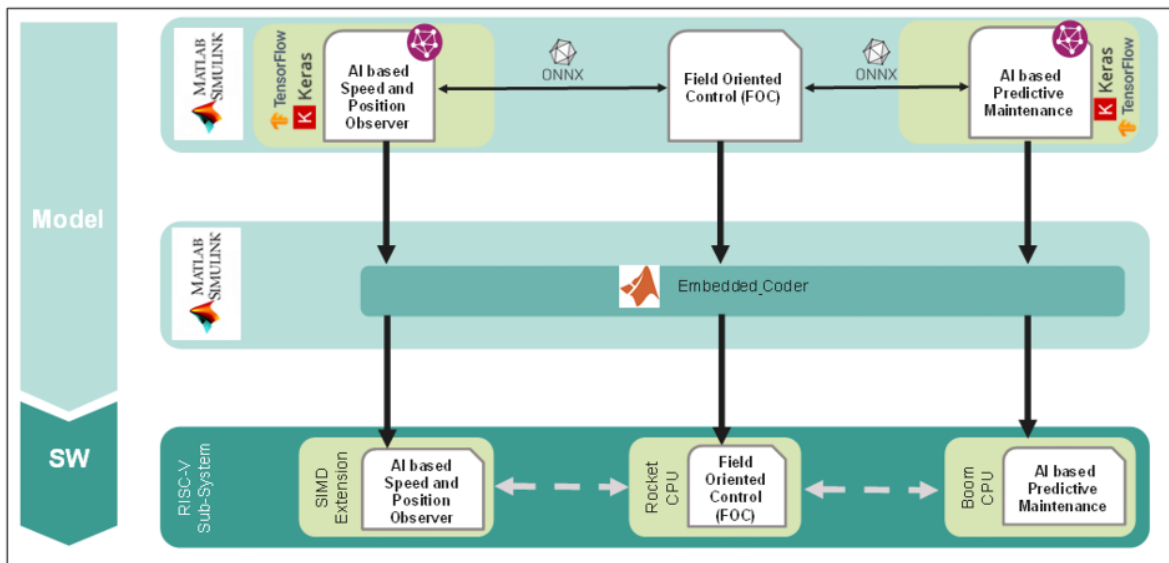


Abbildung 8: Modellbasierte Entwicklungs-Methodik (Quelle: Infineon)

Die obige Abbildung zeigt die Abläufe im Rahmen der modellbasierten Entwicklung und dem Zusammenwirken von AI- und Motor-Control-Algorithmen. Beginnend mit dem AI-based Speed and Position Observer (Simulink) und den Programmier-Tools Tensor Flow und Keras, über die Field Oriented Control (FOC), also die Motor-Steuerung, generiert das Embedded Coder Tool den geforderten C-Code und macht die Lösung auch auf weiteren Subsystemen ablauffähig. Das Infineon-Projektteam hat diese Vorgehensweise für den Einsatz auf Aurix-Mikrocontrollern umgesetzt. Es ist geplant, die in Simulink erstellte Motorregelung mit der RISC V-Hardware kommunizieren zu lassen, um darauf das neuronale Netz zu berechnen. Die Entwickler generierten das Motor-Control Simulink-Modell und stellten die Kommunikation mit der Ziel-Hardware her.

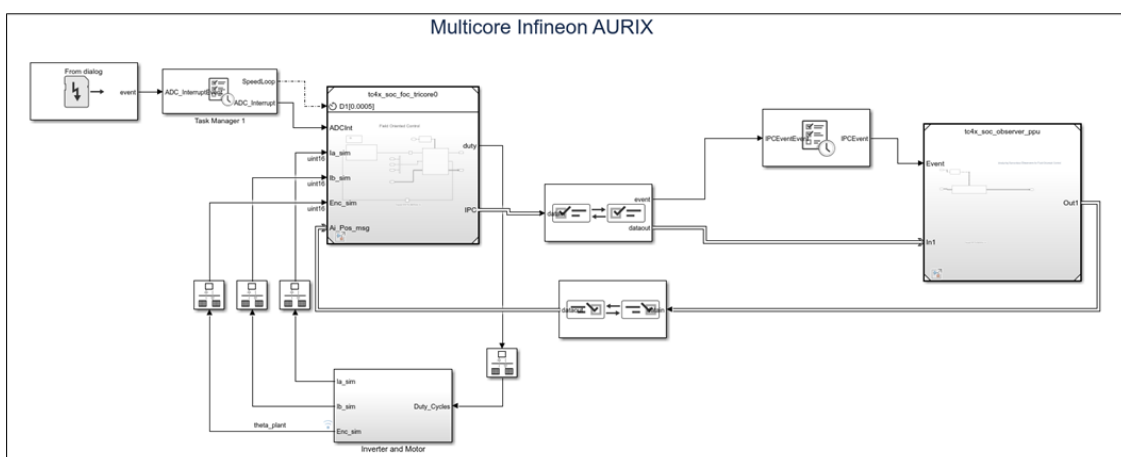


Abbildung 9: Logische Architektur der AI-basierten Motorsteuerung (Quelle: Infineon)

Bei der Umsetzung kam das Embedded Coder Tool von Mathworks zum Einsatz. Es erzeugt lesbaren C-Code, der den Projekt-Mitarbeitern die Steuerung der generierten Funktionen, Dateien und Daten ermöglichte und die Integration mit den Datentypen erleichterte. Embedded Coder unterstützte außerdem die Einbindung von Drittanbieter-Entwicklungstools

zur Erstellung einsatzfertiger Executables für Embedded-Systeme. Der generierte Code ist plattformunabhängig und dadurch auch auf RISC V-Prozessoren ausführbar. Der Prozessor wurde vom Team zunächst als FPGA emuliert, das sich in Versuchen gleichartig verhielt wie die „reale“ Lösung. Die Hardware der Motor-Control-Anwendung wurde simuliert. Die Motorregelung kommunizierte mit der RISC V-Hardware, auf der das neuronale Netz (NN) berechnet wurde. Anschließend übertrugen die Entwickler das NN in die Simulation, um zu untersuchen, ob der Motor sich basierend auf den Berechnungen präzise steuern lässt.

Gemäß den Vorgaben für Arbeitspaket 6.7 integrierte das Infineon-Team auf Softwareebene anwendungsspezifische Überwachungs- und Sicherheitsfunktionen mit Fokus auf Expertise in den Bereichen Motorkontrolle und Bewegungssteuerung. Zu den typischen Funktionen zählten die Temperaturüberwachung, Über- und Unterspannungsüberwachung sowie Überstromüberwachung. Wert legten die Entwickler auf die Möglichkeit, Warn- und Maximalschwellen konfigurieren zu können. Auch das Festlegen von Reaktionen, die im Falle des Eintretens eines Ereignisses automatisch ausgeführt werden, gehörte zu den Vorgaben. Hierzu zählten zum Beispiel das Setzen eines Warnsignals, ein kurzzeitiger Motorstopp oder das komplette Abschalten des Motors. Zu den weiteren Sicherheitsfunktionen, die das Projekt-Team betrachtete, gehörten unter anderem die kontinuierliche Überprüfung der Positions-Feedbacksignale auf Integrität und Plausibilität. Dazu zählte je nach Art des Positionsfeedbacks die Analyse von Zustandsübergängen. Auch die Geschwindigkeitsberechnung und -überwachung sowie der Vergleich redundanter Positionsgeber auf Gleichlauf bzw. Abweichung zählten hierzu. Als weitere Features adressierte das Infineon-Team die Überwachung der Stopp-Signale zur Sicherheitsabschaltung und die daraufhin ausgeführten Aktionen, um das System in einen vorab definierten Zustand zu bringen. Die Aktivitäten im AP6.7 wurden über den Projektverlauf weitergeführt. Details hierzu sind am Ende des Reports unter dem Punkt AP6.7 aufgeführt.

Den Schwerpunkt der Infineon-Aktivitäten im AP6.8 bildeten die Bereiche Predictive Maintenance (vorausschauende Wartung) und Condition Monitoring (Zustandsüberwachung). Predictive Maintenance versetzt Unternehmen in die Lage, Systemausfälle vorherzusehen und somit zu vermeiden. Die Aktivitäten im AP6.8 wurden über den Projektverlauf weitergeführt. Details hierzu sind am Ende des Reports unter dem Punkt AP6.8 aufgeführt.

Arbeitspaket 6.9 (AP6.9) „Testfunktionen für Qualifikation und Validierung“

Arbeitspaket 6.9 hatte „Testfunktionen für Qualifikation und Validierung“ zum Thema. Zur Qualifizierung und Validierung der Demonstratoren sowie der digitalen Funktionsblöcke wurden von den Infineon-Ingenieuren automatische Testabläufe entworfen, die einen automatischen Systemintegrationstest ermöglichten. Zugleich ließ sich der Einfluss von Änderungen in bestimmten Funktionsblöcken auf andere Funktionalitäten feststellen. Ebenso wie das Hardware-IP-Portfolio modular gestaltet wurde, achtete das Projektteam auch hier auf Modularität, um einen hohen Grad an Nachnutzbarkeit zu erreichen. Die Automatisierung ermöglichte schnelle Durchlaufzeiten und eine gute Wiederverwertung in späteren Projekten.

Dabei waren eine erweiterte Testumgebung und Testbenches hilfreich, mit denen sich Teile des Systems stimulieren und prüfen ließen.

Arbeitspaket 6.10 (AP6.10) „Anwenderwerkzeuge“

Im Arbeitspaket 6.10 „Anwenderwerkzeuge“ stand die Benutzbarkeit der geplanten Entwicklungsplattform für eingebettete Microcontroller obenan. Der Erfolg hing erheblich von ihrer Zugänglichkeit und Handhabbarkeit ab. Benutzbarkeit definierte sich in diesem Zusammenhang durch Ersparnis an Aufwand und Zeit, aber auch durch Vereinfachung der komplexen technischen Systeme und ihre Wiederverwendbarkeit. Um dies zu erreichen, nutzte das Infineon-Team bei der Entwicklung der Plattform Qualifizierungs- und Validierungs-Tools. Zum Ende des Projekts lagen Werkzeuge zur Benutzung der Entwicklungsplattform für eingebettete Prozessoren und Microcontroller vor, die es Benutzern erlauben, angepasste Systeme zu konfigurieren und zu verifizieren. Die projektbegleitenden Entwicklungswerkzeuge wurden mit Testfunktionen für Qualifikation und Validierung (AP6.6) gekoppelt. Die Infineon-Entwickler entwarfen zusätzlich zu Treibern und Beispielscodes Werkzeuge, die die Konfiguration der Peripherieblöcke erleichtern und die intuitive Inbetriebnahme der Demonstratoren und Produkte ermöglichten.

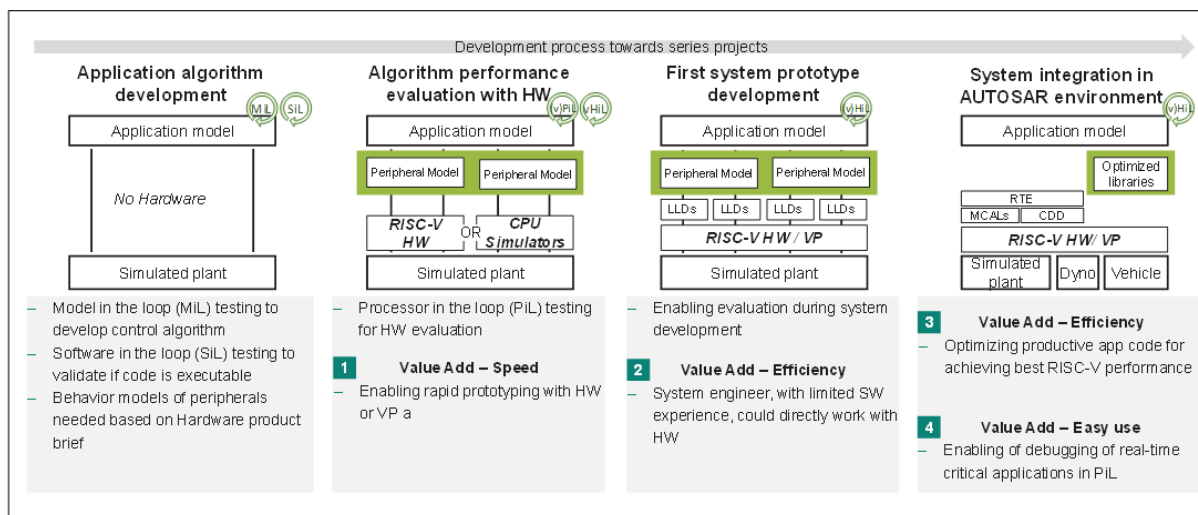


Abbildung 10: Schematische Darstellung des Infineon-Entwicklungsprozesses (Quelle: Infineon)

Die obige Abbildung zeigt schematisch den vom Infineon-Projektteam gestalteten Entwicklungsprozess. Beginnend mit der Entwicklung der Anwendungsalgorithmen, über die Bewertung der Algorithmenleistung und Erstellung eines ersten Systemprototyps zur Integration in die AUTOSAR-Umgebung.

Die Arbeiten des KI-IoT-Projekts lagen für das Infineon-Projektteam im letzten Berichtszeitraum (Projektmonate 31 bis 36) schwerpunktmäßig auf dem Arbeitspaket 6. Der Fokus lag hier auf den Themen „Überwachung und Sicherheitsfunktionen“ (AP6.7) sowie „Predictive Maintenance“ (AP6.8).

Arbeitspaket 6.7 (AP6.7) „Überwachung und Sicherheitsfunktionen“

Im Rahmen von AP6.7 „Überwachung und Sicherheitsfunktionen“ arbeitete das Infineon Projekt-Team eng mit dem Fraunhofer-Institut für Kognitive Systeme IKS zusammen. Gemeinsam führten die Partner eine Sicherheitsbewertung der Funktionalität (SOTIF) durch. Das Ziel war, im Rahmen der technischen Produktsicherheit die Widerstandsfähigkeit und Leistungsfähigkeit der Software-Entwicklungswerkzeuge zu analysieren.

Die Integration von künstlicher Intelligenz (KI) in Automobilsysteme ist mit komplexen Herausforderungen in Bezug auf Sicherheit, Zuverlässigkeit und Vertrauenswürdigkeit verbunden. Um diesen Anforderungen wirkungsvoll zu begegnen, sind robuste Zuverlässigkeitsstrategien erforderlich. Sie stellen sicher, dass KI-basierte Komponenten unter den verschiedenen Bedingungen wie vorgesehen funktionieren. Die kürzlich eingeführte Norm ISO/PAS 8800 bietet Richtlinien, die auf die Sicherung von KI-Systemen zugeschnitten sind und einen Rahmen für die Bewältigung der Risiken und Unsicherheiten bieten. Im Rahmen eines Assessments untersuchte das gemeinsame Projektteam eine KI-basierte Motorsteuerungseinheit (AI-MC) zusammen mit einem KI-basierten Positionssensor (AI-PS). Bei dem Positionssensor handelte es sich um ein neuronales Netz (NN), das Ströme und Spannungen als Eingaben verarbeitete, um die Position der Motorwelle zu schätzen.

Als weiterer Aspekt bei der Bearbeitung der AP6-Anforderungen wurde vom Infineon-Team die Laufzeitüberwachung adressiert. Sie spielt eine entscheidende Rolle, wenn es darum geht, die Ausfallsicherheit eines KI-Systems während des Betriebs zu gewährleisten, d.h. wenn das System auf Bedingungen stößt, die über seinen nominellen Arbeitsbereich hinausgehen, einschließlich der Widerstandsfähigkeit. Während die Robustheit in der Entwurfsphase von KI-Komponenten berücksichtigt wird, bieten Laufzeiten ein dynamisches Sicherheitsnetz, um Abweichungen in Echtzeit zu erkennen und darauf in Echtzeit zu reagieren. Mechanismen, wie z. B. die OOD-Erkennung (Out-of-Distribution), sind entscheidend für Szenarien, in denen die KI-Komponente zu versagen beginnt. Sie verhindert Break-downs auf Systemebene.

Die Laufzeitüberwachung umfasste die kontinuierliche Verfolgung des Verhaltens des eingesetzten KI-Modells, um sicherzustellen, dass es zuverlässig arbeitete und genaue Vorhersagen lieferte. Dieser Prozess ist entscheidend für die Erkennung potenzieller Probleme, wie z. B. Unsicherheiten bei Vorhersagen, OOD-Eingaben oder Leistungsverschlechterungen aufgrund von Verschiebungen der Datenmuster im Laufe der Zeit. Für die Laufzeitüberwachung wurden zwei sich ergänzende Komponenten verwendet: OOD-Erkennung und Uncertainty Quantification (UQ). Die Erkennung von OOD-Fehlern konzentrierte sich auf die Erkennung von Eingaben, die von der Datenverteilung während des Trainings abwichen. Solche Eingaben können von Umweltveränderungen, Sensorrauschen oder Betriebsverschiebungen stammen und führen oft zu unzuverlässigen Vorhersagen.

Während die OOD-Erkennung anomale Eingaben identifizierte, bewertete die Uncertainty Quantification (UQ) das Vertrauen in die Modell-Vorhersagen. Eine hohe Unsicherheit wies entweder auf OOD-Eingaben oder schlecht verstandene Regionen des Eingaberaums hin. Die Architektur der Laufzeitüberwachung für den KI-basierten Positionssensor (AI-PS) integrierte OOD-Erkennungsmethoden mit UQ, um eine robuste Entscheidungsfindung zu unterstützen. Wenn OOD-Eingänge oder hohe Unsicherheiten erkannt wurden, reagierte das System

entsprechend, z. B. Durch Aufrufen eines Backup-Sensors oder durch sicheres Abschalten des Motors. Dies entsprechend dem für das AI-MC-System geforderten Sicherheitskonzept.

Ähnlich wie die Sicherheitsanforderungen an die Widerstandsfähigkeit wurden eine Reihe von Sicherheitsanforderungen in Bezug auf Maßnahmen zur Betriebszeit definiert. OOD-Daten beschrieben Eingaben, auf die ein KI-Modell während der Inferenz oder des Testens traf und die außerhalb der Verteilung der Trainingsdaten lagen. Solche Daten führten zu unzuverlässigen oder ungenauen Vorhersagen, da das Modell nicht in der Lage war, über die erlernten Muster hinaus zu verallgemeinern. Dies konnte wiederum zu schlechter Leistung oder sogar zum Versagen führen. Die OOD-Erkennung bot eine Möglichkeit, solche Eingaben zu identifizieren, bevor sie den Systembetrieb beeinträchtigten. Die für die OOD-Erkennung in Betracht gezogenen Methoden hatten einen gemeinsamen Ansatz: Sie erzeugten jeweils eine numerische Punktzahl, die die Wahrscheinlichkeit darstellte, dass eine Eingabe zur gleichen Verteilung gehörte wie die Trainingsdaten. Dieser Wert wurde dann mit einem Schwellenwert verglichen, um zu entscheiden, ob es sich bei der Eingabe um ID oder OOD handelte. Wichtig war, dass der Schwellenwert so gewählt wurde, dass die Daten aus dem Testsatz, bei denen davon ausgegangen wurde, dass sie zur Verteilung gehörten, nicht zu viele falsch-positive Ergebnisse lieferten.

Arbeitspaket 6.8 (AP6.8) „Predictive Maintenance“

„Predictive Maintenance“ war der Fokus von AP6.8. Predictive Maintenance versetzt Unternehmen in die Lage, Systemausfälle vorherzusehen und so zu vermeiden. Das Ziel des Infineon-Projektteams war es herauszufinden, welche Rolle Mikrocontroller hierbei übernehmen können. Das Projektteam untersuchte zunächst die Arbeitsschritte der klassischen Wartung. Bei der periodischen Wartung überprüft ein Mitarbeiter gemäß einem Wartungsplan die entsprechenden Bauteile und deren Zustand, um bei Bedarf Verschleißteile auszutauschen. Um einen Schaden zwischen Wartungsintervallen zu vermeiden, stellt Predictive Maintenance einen wirkungsvollen Ansatz dar. Der Wartungsmitarbeiter schätzt den Zustand der Komponenten mittels Condition Monitoring, bei dem die aktuellen Anlagen- und Maschinenzustände und deren Abnutzung ausschlaggebend sind. Predictive Maintenance prognostiziert, wann eine Wartung notwendig ist. Somit kann ein potenzieller Maschinenausfall auch außerhalb fester Wartungszyklen erkannt und vermieden werden.

Predictive Maintenance ist gleichzusetzen mit vorausschauender Instandhaltung. Bei dieser Form der Instandhaltung geht es darum, Systemausfälle vorherzusehen und zu vermeiden. Das Infineon-Team entwickelte ein System, mit dem sich verschiedene Parameter kontinuierlich beobachten und auswerten lassen. Sobald das System Funktionsunregelmäßigkeiten oder Abnutzungserscheinungen erkennt, werden Wartungsarbeiten oder Reparaturen in den Regelbetrieb der Anlage oder Maschine eingeplant. Bevorstehende Systemausfälle werden vorausschauend erkannt und abgewendet. Bei Preventive Maintenance geht es darum, Maßnahmen zu ergreifen, die unerwartete Ausfälle minimieren und die Planbarkeit der Instandhaltung maximieren. Kennzeichnend für diese Strategie sind regelmäßige Prüfabstände wie festgelegte Wartungs-, Tausch- und Reparaturintervalle. Durch festgelegte Zyklen wird Verschleiß früh erkannt. Eventuelle Ausfälle werden so auch außerhalb der üblichen Prüfintervalle des Preventive Maintenance erkannt und konkret vorhersagbar.

Predictive Maintenance wird durch leistungsfähige Systeme für Condition Monitoring möglich. Das Infineon-Projektteam setzte hierzu ein System aus Mikrocontrollern, integrierten Leistungs- und Konnektivitätsschaltungen (ICs) und Sensoren auf. Dieses System überwachte fortlaufend die Maschinen und Anlagensysteme und machte durch den ständigen Abgleich zwischen Soll- und Ist-Werten Abweichungen von Normwerten erkennbar. Das Wartungspersonal bekam schnell einen Überblick über mögliche Verschleiß- und Schadensgründe. Wartungs- und Reparaturbedarfe wurden planbar und kosteneffizienter.

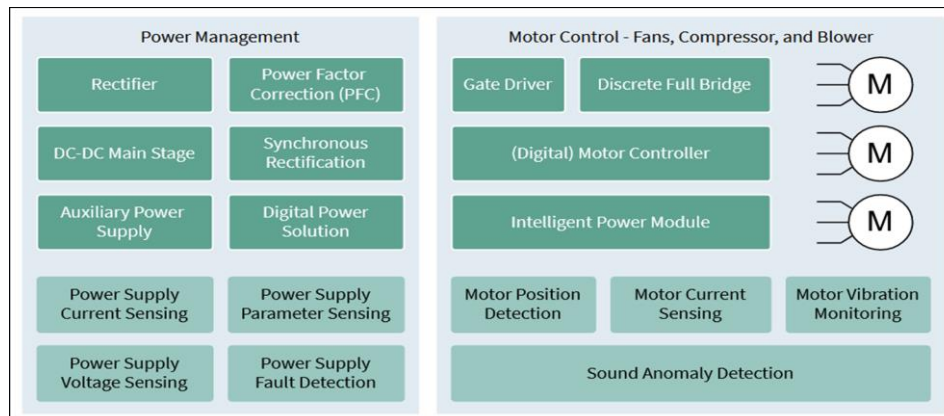


Abbildung 11: Zustandsüberwachung kritischer Parameter (Quelle: Infineon)

Mit Condition Monitoring und den damit generierten Daten wurden Grundsteine für Predictive Maintenance gelegt. Für die Anforderungen im Rahmen des Arbeitspakets 6.8 testete das Infineon-Team mit Evaluation Kits, welche Sensorsysteme für den jeweiligen Einsatzbereich optimal geeignet sind und sich mit applikationsspezifischer Software kombinieren lassen, um eine umfassende Datenquelle für Predictive Maintenance zu erlangen. Von der Messung bis zur Datenbewertung durchlief die Predictive-Maintenance Lösung mehrere Schritte. Das Aufgabenspektrum begann mit der Erhebung der Daten durch die Sensoren, gegebenenfalls auch durch Zusammenführen von Sensorinformationen (Sensor Fusion). Sobald man die Daten mehrerer Aggregate erfasst hatte, liefen diese in einem IoT-Sensorknoten zusammen und wurden dort verarbeitet. Je nach Maschine oder Anlage wurde entschieden, die Auswertung der Predictive-Maintenance-Software direkt am jeweiligen Aggregat durchzuführen oder die Daten zunächst weiterzuleiten. Im ersten Fall konnten die Daten „at-the-edge“ verarbeitet werden, bevor sie an ein zentrales System, in diesem Fall eine Cloud-Plattform, weitergeleitet und dort ausgewertet wurden.

Vom Microcontroller wurden die Sensordaten über Wi-Fi- oder Bluetooth-Schnittstellen an ein zentrales Management-System oder über die Cloud an externe Dienste weitergeleitet. Dort wurden sie mit KI-Algorithmen ausgewertet. Die vom Infineon-Team entwickelte Software lief komplett autonom „at-the-edge“. Zum Beispiel konnte eine Maschine mit einem zentralen System oder mittels Cloud mit anderen Maschinen kommunizieren. KI-Algorithmen verglichen den Ist-Zustand mit dem Soll-Zustand der zu überwachenden Geräte. Etwaige Anomalien wurden erkannt und gegebenenfalls ein Alarm ausgelöst. Mit Predictive-Maintenance-KI wurden die relevanten Daten mit einem Modell aus Machine Learning und Big Data ausgewertet. Daten wie Luftzusammensetzungen, Vibrationen oder Temperaturen

wurden genutzt, um Zusammenhänge zwischen Werten zu ermitteln. Die KI-Lösung erlaubte es, eine Vorhersage zu treffen, ob und wann ein Defekt eintreten würde. So war es möglich, den Systemzustand präzise vorherzusagen. Der Einsatz von KI-Beschleunigern forcierte die Mustererkennung (Prüfmuster). Mustererkennungs-Algorithmen wurden aufgrund ihrer Komplexität auf der MC-Plattform abgebildet. Erst im Anschluss daran erfolgte die Einbindung der Beschleuniger, um Rechenzeiten zu reduzieren.

2. Fazit

Die im KI-IoT-Projekt erzielten Ergebnisse waren für alle beteiligten Partner sehr zufriedenstellend. Hervorzuheben war auch die Kooperation der beteiligten Projektpartner. Auf Grundlage dieser Resultate wurde eine Adaption der AI/ML-Lösung zur Implementierung auf dem RISC-V System durchgeführt. Die Integration wurde von den Partnern teilweise in der Verlängerung des Projektes durchgeführt und durch Infineon unterstützt. Das KI-IoT-Projekt war für die Bereiche Automobilelektronik, Antriebstechnik und Industrieautomatisierung von hoher Relevanz. Die erforschte KI-basierte Motorsteuerung wird nicht nur in nachfolgende Produktentwicklungen, sondern auch in neue R&D-Projekte bei Infineon einfließen. Daraus ergibt sich beispielsweise die Möglichkeit, kontinuierlich die wachsenden Anforderungen an elektrische Antriebe bezüglich höchster Zuverlässigkeit mit einer kleinstmöglichen Time-to-Market und innovativen Produkten zu adressieren. Im Bereich der Industrieautomatisierung strebt Infineon an, vergleichbare Produkte und Systeme unter anderem auch im Robotik-Sektor zu platzieren.

3. Abbildungsverzeichnis

Abbildung 1: Arbeitsplan und Berichtszeiträume der Arbeitspakete (Quelle: Infineon).....	4
Abbildung 2: Sensorlose Motorregelung (Quelle: Infineon).....	7
Abbildung 3: SIMD-Befehlssatzerweiterung des RISC-V-Cores (Quelle: UzL/Infineon).....	8
Abbildung 4: AI-basierte sensorlose feldorientierte Motorregelung (FOC) (Quelle: Infineon).....	13
Abbildung 5: Schaltdiagramm einer Extended MLP Motorregelung (Quelle: Infineon).....	13
Abbildung 6: AI-basierter Positionssensor für Field Oriented Control (FOC) (Quelle: Infineon).....	14
Abbildung 7: Mittlerer absoluter Fehler (MAE) über Eingabehistorie (Quelle: Infineon)	15
Abbildung 8: Modellbasierte Entwicklungs-Methodik (Quelle: Infineon)	16
Abbildung 9: Logische Architektur der AI-basierten Motorsteuerung (Quelle: Infineon).....	16
Abbildung 10: Schematische Darstellung des Infineon-Entwicklungsprozesses (Quelle: Infineon).....	18
Abbildung 11: Zustandsüberwachung kritischer Parameter (Quelle: Infineon)	21




KI-IoT

Holistische Open-Source-Plattform
für eingebettete Systems-on-Chip

Projekt **KI-IoT**

Schlussbericht gemäß NKBF 2017 – Teil I TIB
Kurzbericht
der Infineon Technologies AG

Zuwendungsempfänger: Infineon Technologies AG 81726 München		Förderkennzeichen: 16ME0351
Vorhabenbezeichnung: Entwicklungsplattform für vertrauenswürdige IoT-Mikrochips mit innovativem KI-Co- Prozessor – ZUSE-KI-IoT		
Teilvorhaben: Beitrag zur Holistischen Open-Source-Plattform für eingebettete Sys- tems-on-Chip mit Fokus auf KI-basierte Motorsteuerungen in Automotive		
Laufzeit des Vorhabens / Berichtszeitraum: 01.07.2021 bis 30.06.2024		
Berichtsdatum 20.12.2024		
Vorgelegt beim Projektträger: VDI/VDE Innovation + Technik GmbH Steinplatz 1 10623 Berlin		
Vorgelegt durch: Infineon Technologies AG Dr. Stephan Eder - SFM RTF CFA Am Campeon 1-15 85579 Neubiberg		

Ziel des KI-IoT-Vorhabens war die Entwicklung und quelloffene Bereitstellung einer modularen Plattform für den Entwurf von Systems-on-Chip (SoC) auf Basis der RISC-V-Architektur mit eingebetteten nichtflüchtigen Speichern (NVM). Der Open-Source-Hardware-Ansatz für digitale Kernkomponenten sowie Lizenzmodelle analoger und proprietärer Funktionsblöcke soll die Einstiegshürden - insbesondere in KMUs - für anwendungsspezifische Prozessoren deutlich senken. Das KI-IoT-Projekt begann am 01. Juli 2021 und endete am 30. Juni 2024. Es gliederte sich in zehn in Wechselbeziehung stehende Arbeitspakete. Während der 36-monatigen Laufzeit lagen die Schwerpunkte der Infineon-Aktivitäten auf den Arbeitspaketen „Entwicklung Testumgebung“, „Digitalentwurf“ sowie „Firmware und Inbetriebnahmewerkzeuge“.

Unter der Bezeichnung „Entwicklung Testumgebung“ waren Aktivitäten wie die Evaluierung von Open-Source IP-Blöcken sowie der Entwurf von Entwicklungsboards auf Basis von Field Programmable Gate Arrays (FPGAs) zusammengefasst. An das FPGA schloss das Infineon-Projektteam die Peripherie zur Motorregelung und Lageerfassung an und es verband die Schnittstellencontroller mit dem FPGA. Zur Inbetriebnahme stimmten sie Hardwareentwurf, Digitaldesign und Softwareentwicklung eng aufeinander ab.

Der „Digitalentwurf“ bestand aus mehreren Teilarbeitspaketen. Ein Arbeitspaket adressierte die Standardperipherie. Microcontroller benötigen Standard-Peripherieblöcke zur Kommunikation und Datenerfassung. Sie bilden das Kernstück eines Funktionsmoduls, das externe Peripherieeinheiten bedient. In einem weiteren Arbeitspaket analysierte das Infineon-Projektteam Lösungen, deren Einsatz die Positions- und Drehzahlerfassung von E-Motoren optimiert. Die Team-Mitarbeiter fokussierten sich zudem auf das Thema „Kompensation von Nichtlinearitäten in Servoantrieben“. Bestimmte Komponenten eines Servoantriebs weisen fertigungs- und technologiebedingte Nichtlinearitäten auf. Zur Kompensation der Nichtlinearität entwarf das Projektteam selbststellende Kompensationsverfahren und implementierte sie in Hardware. Die Themen Firmware und Inbetriebnahmewerkzeuge standen in einem weiteren Arbeitspaket obenan. Das Infineon-Projektteam entwickelte Treiber und Application Programming Interfaces (APIs) für Peripherie-Einheiten der Prozessorplattform. Die Arbeiten umfassten ebenso die Bereitstellung von Software für Open-Source-Peripherieblöcke.

Einen Schwerpunkt bildete auch die Entwicklung und Software-Implementierung einer sensorlosen Motorregelung auf RISC-V-Basis und die Implementierung in Neural Networks (MLP, LSTM etc.). Bei der Softwarerealisierung war die Flexibilität in der Gestaltung und Auslegung eines Systems innerhalb des Baukastens für eingebettete Microcontroller das Ziel. Sie erlaubte den Entwicklern die Realisierung anwendungsspezifischer Systeme, bei denen nicht höchste Leistungsfähigkeit der Hardware im Vordergrund stand, sondern stattdessen auf möglichst kleine Siliziumflächen und niedrige Kosten Wert gelegt wurde. Eine andere Forderung an die Infineon-Entwickler lautete „Implementierung und Integration von CPU-Kerneinheiten“. Dies betraf vor allem die Verbindung der On- und Off-Chip-Komponenten. Über einen breitbandigen Bus band das Team die On-chip-Peripherie an die CPU an.

Im Arbeitspaket 5.4 „Lage- und Drehzahlerfassung“ befassten sich die Infineon-Entwickler mit industrietypischen Feedbacksystemen, die für die Servoregelung durch digitale Funktionsblöcke ausgewertet werden. Neben klassischen Inkrementalgebern sind digitale Hallsignale bei permanent erregten Synchronmotoren (PMS: Permanent Magnet Synchronous Machines) verbreitet. Durch den Einsatz von modellbasierten

Beobachterstrukturen lässt sich die Positions- und insbesondere die Drehzahlerfassung deutlich verbessern. Zusätzlich können nichtlineare und systematische Abbildungsfehler der Positionsgeber korrigiert werden. Im Rahmen des KI-IoT-Projekts erprobten die Entwickler beobachterbasierte Drehzahlerfassungsmethoden für den Einsatz in den Demonstratoren.

Die Arbeiten im AP5.6 waren unter dem Thema „Entwicklung sensorloser Motorregelung“ zusammengefasst. Für die klassische Servoregelung eines elektrischen Antriebs sind mehrere digitale Funktionsblöcke erforderlich. Hierzu gehören u.a. ADC-Auswertung, Pulsweitenmodulation (PWM), Motorregelung, Bahnplanung und Ablaufsteuerung. Ein Ziel des Forschungsvorhabens war die Umsetzung von Funktionen zur Bewegungsführung und -planung (Motion Control) und zur Motorregelung (Motor Control).

Im Arbeitspaket 6.10 „Anwenderwerkzeuge“ stand die Benutzbarkeit der geplanten Entwicklungsplattform für eingebettete Microcontroller obenan. Um dies zu erreichen, nutzte das Infineon-Team bei der Entwicklung der Plattform Qualifizierungs- und Validierungs-Tools. Zum Ende des Projekts lagen Werkzeuge zur Benutzung der Entwicklungsplattform vor, die es Benutzern erlauben, angepasste Systeme zu konfigurieren und zu verifizieren.

Die Arbeiten des KI-IoT-Projekts lagen für das Infineon-Projektteam im letzten Berichtszeitraum (Projektmonate 31 bis 36) schwerpunktmäßig auf dem Arbeitspaket 6. Der Fokus lag hier auf den Themen „Überwachung und Sicherheitsfunktionen“ sowie „Predictive Maintenance“. Im Rahmen des APs „Überwachung und Sicherheitsfunktionen“ arbeitete das Infineon Projekt-Team eng mit dem Fraunhofer-Institut für Kognitive Systeme IKS zusammen. Gemeinsam führten die Partner eine Sicherheitsbewertung der Funktionalität (SOTIF) durch. Das Ziel war, im Rahmen der technischen Produktsicherheit die Widerstandsfähigkeit und Leistungsfähigkeit der Software-Entwicklungswerkzeuge zu analysieren.

„Predictive Maintenance“ war der Fokus von AP6.8. Predictive Maintenance versetzt Unternehmen in die Lage, Systemausfälle vorherzusehen und so zu vermeiden. Das Ziel des Infineon-Projektteams war es herauszufinden, welche Rolle Mikrocontroller hierbei übernehmen können. Das Projektteam untersuchte zunächst die Arbeitsschritte der klassischen Wartung. Bei der periodischen Wartung überprüft ein Mitarbeiter gemäß einem Wartungsplan die entsprechenden Bauteile und deren Zustand, um bei Bedarf Verschleißteile auszutauschen. Um einen Schaden zwischen Wartungsintervallen zu vermeiden, stellt Predictive Maintenance einen wirkungsvollen Ansatz dar. Der Wartungsmitarbeiter schätzt den Zustand der Komponenten mittels Condition Monitoring, bei dem die aktuellen Anlagen- und Maschinenzustände und deren Abnutzung ausschlaggebend sind. Der Mitarbeiter prognostiziert, wann eine Wartung notwendig ist. Somit kann ein potenzieller Maschinenausfall auch außerhalb fester Wartungszyklen erkannt und vermieden werden.

Die im KI-IoT-Projekt erzielten Ergebnisse waren für alle beteiligten Partner sehr zufriedenstellend. Hervorzuheben war auch die Kooperation der beteiligten Projektpartner. Auf Grundlage dieser Resultate wurde eine Adaption der AI/ML-Lösung zur Implementierung auf dem RISC-V System durchgeführt. Die Integration wurde von den Partnern teilweise in der Verlängerung des Projektes durchgeführt und durch Infineon unterstützt. Das KI-IoT-Projekt hatte für die Bereiche Automobilelektronik, Antriebstechnik und Industrieautomatisierung hohe Relevanz. Die erforschte KI-basierte Motorsteuerung wird nicht nur in nachfolgende Produktentwicklungen, sondern auch in neue R&D-Projekte bei Infineon einfließen. Daraus

ergibt sich beispielsweise die Möglichkeit, kontinuierlich die wachsenden Anforderungen an elektrische Antriebe bezüglich höchster Zuverlässigkeit mit einer kleinstmöglichen Time-to-Market und innovativen Produkten zu adressieren. Im Bereich der Industrieautomatisierung strebt Infineon an, vergleichbare Produkte und Systeme unter anderem auch im Robotik-Sektor zu platzieren.