Funktionale Sicherheit in Automotive und Avionik: Ein Staffellauf

Andreas Schwierz[1] Georg Seifert[1] Sebastian Hiergeist[1]

Abstract: Der nachfolgende Bericht geht auf die gemeinsamen Interessen von sicherheitskritischen Systemen aus der Luftfahrt- und der Automobilbranche ein. Hierbei wird dargelegt, dass die Software-Funktionalität stark von der eingesetzten Hardware abhängig ist und Auswirkungen auf die gewünschte Sicherheit hat. In diesem Bereich können beide Branchen voneinander profitieren. Die Luftfahrt hat historisch gesehen schon früh angefangen, systematisch funktionale Sicherheit zu standardisieren, wohingegen die Automobilbranche seit 2011 nachzieht und mit ihrer großen Marktmacht auf die Hardwarehersteller einwirken kann. Hieraus könnte auch die Luftfahrtindustrie ihren Nutzen ziehen.

Keywords: Luftfahrt, Automobil, Funktionale Sicherheit, Echtzeitsysteme, WCET, Redundanzsystem, Interferenz, Zugriffskollisionen

1 Einleitung

Luftfahrtzeug- und Automobilhersteller teilen beide das Interesse, sichere Beförderungsmittel bereitzustellen. Getrieben durch den Wandel von mechanischen zu elektrischen/-elektronischen (e/e) Systemen begann in der Luftfahrt ab den 1980ern[2] eine strukturierte Auseinandersetzung mit funktionaler Sicherheit. Die Herausforderung war eine Beibehaltung bzw. Steigerung der Sicherheitsansprüche um ein Fail-Operational Verhalten garantieren zu können. Erreicht wurde dies durch die Entwicklung von Redundanzarchitekturen auf Systemebene. Verwendung fanden hierbei Mikroprozessoren (Micro Processor Unit, MPUs), die mit proprietärer Erweiterungen (ASICs, FPGAs, usw.) um die entsprechende Redundanzfunktionalität erweitert wurden. Im Vordergrund stand hier die Sicherheit des Gesamtsystems, während andere Faktoren wie Gewicht, Größe, Energieverbrauch und Kosten einen untergeordneten Stellenwert einnahmen.

Die Automobilbranche hat erst mit der Veröffentlichung des ISO 26262 [Te09] im Jahr 2011 ein Äquivalent zu den Avionik-Standards definiert. Bedingt durch die Anforderungen aus der Automobilbranche lag der Entwicklung des Standards eine andere Anforderungsbasis zugrunde. Der Hauptunterschied besteht darin, dass ein Fail-Safe Verhalten für solche Systeme ausreichend ist. Die oben genannten Faktoren wie Gewicht, Größe, Energieverbrauch und Kosten haben bei diesen Systemen einen ungleich höheren Stellenwert. Um diese Anforderungen erfüllen zu können wurde seitens der Chip-Hersteller viel

[1] Technische Hochschule Ingolstadt, Zentrum für Angewandte Forschung, Paradeplatz 13, 85049 Ingolstadt, Vorname.Nachname@thi.de

[2] Im Jahr 1982 wurde der Standard DO-178[Sp82] veröffentlicht.

Entwicklungs- und Forschungsaufwand betrieben, um die geforderten Safety-Aspekte im Rahmen des ISO 26262 zu erfüllen.

Dem historisch bedingten Erfahrungsvorsprung zum Trotz konnte die Luftfahrtindustrie hingegen keinen Einfluss auf die Entwicklung sicherheitskritischer und gleichzeitig hochintegrierter Hardware-Komponenten ausüben[FK06]. In sicherheitskritischen Avionik-Systemen muss dies durch den Einsatz von bewährten MPUs, in Kombination mit einer Redundanz auf Systemebene, gelöst werden. Zur Steigerung der Integrationsdichte wurde in der zivilen Luftfahrt der Integrated Modular Avionic (IMA) Ansatz entwickelt, der erstmals im Airbus A380 zum Einsatz kam. Diese Architektur ist auf die Verwendung in großen, komplexen Systemen ausgelegt und für den Einsatz in sicherheitskritischen Avionik-Systemen ungeeignet. In Anbetracht der anvisierten Einsatzszenarien von unbemannten Flugkörpern[Vo13] müssen deswegen funktional hochintegrierte Mikrocontroller (Microcontroller Unit, MCUs) verwendet werden.

Seitens der Automotive Mikrocontroller (Automotive Microcontroller Unit, AMCUs)-Hersteller standen hier domänenspezifischen Entwicklungsanforderungen im Vordergrund. Damit AMCUs in künftigen Avionik-Systemen eingesetzt werden können, muss die Zulassbarkeit argumentiert werden. Bei der Entwicklung des Avionik-Systems entstehen Commercial off-the-shelf (COTS)-spezifische herausforderungen, von denen drei in den nachfolgenden Kapiteln aufgegriffen werden.

Kapitel 2 erläutert den Stellenwert der **Qualität** bei der Avionik-Herstellung und stellt die Frage, ob Hardware-Komponenten, entwickelt nach ISO 26262, eine Erleichterung in der Flugzeugzulassung darstellen können. Trotz der umfangreichen Safety-Features innerhalb der AMCU wird die **Zuverlässigkeit** einer einzelnen AMCU nicht ausreichen, um den höchsten Sicherheitsansprüchen gerecht zu werden. Deswegen werden in Kapitel 3 aktuelle Sicherheitsbedenken untersucht und entsprechende Lösungsmöglichkeiten aufgezeigt. Als sinnvolle Lösung erscheint in diesem Zusammenhang die Realisierung eines Redundanznetzwerkes durch AMCU-eigenen Bordmittel. Aufgrund der dadurch stark steigenden Datenlast auf den einzelnen MCU müssen die Einflüsse von Eingabe/Ausgabe (E/A)-Datenflüssen auf die Software-Ausführungszeit detaillierter untersucht werden. Dieser Aspekt wird in Kapitel 4 ausführlich betrachtet.

2 Avionik-Entwicklung: Qualität von komplexen COTS

In der Luftfahrt und Automobilindustrie beschreiben domänenspezifische Standards die Entwicklung von sicherheitskritischer Hardware als Teil eines Systems. In der Luftfahrt ist dies der Standard DO-254[Sp00] – in der Automobilindustrie der ISO 26262[Te09]. Sie sind aus dem Bewusstsein entstanden, dass ein strukturierter Entwicklungsprozess notwendig ist um systematische Fehler bei komplexen Komponenten zu vermeiden. Dies muss das Ziel während der Entwicklung sein, da durch eine umfangreiche Verifikation nicht alle Entwicklungsfehler aufgedeckt werden können.

Diese Erkenntnis gilt für komplexe COTS-Komponenten. Deren Entwicklungsprozess bzw. die daraus resultierende Qualität muss zulassungskonform sein. Das heißt, es muss hierbei

eine qualitative Bewertung über den Entstehungsprozess der Komponente erstellt werden. Das daraus resultierende Ergebnis ist eine Vertrauensaussage über die Integrität bzw. Qualität der COTS-Komponente. Als Bewertungsgrundlage für die Qualität kann dabei der domänenspezifische Standard herangezogen werden. Dieser Vergleich ist gewinnbringender, wenn der COTS-Entwicklungsprozess auf einem verbreiteten Standard beruht. Da die gewonnenen Erkenntnisse auf einem Vergleich zwischen zwei Standards basieren ist das Abstraktionsniveau hoch. Die Wiederverwendung ist dadurch unabhängig von Hersteller und Produkt.

Der AMCU stellt für die Luftfahrtindustrie eine komplexe COTS-Komponente dar und erfüllt die Anforderung: Entwickelt nach einem verbreiteten Standard. Die Zielsetzung des ISO 26262 bzgl. der Entwicklung funktional sicherer Systeme ist eine Ausgangsvoraussetzung für einen Vergleich. Diese Halbleiterprodukte werden in Zukunft weitere sicherheitsrelevante Funktionen übernehmen und die Hersteller sind sich der steigenden Nachfrage bewusst. Mit diesem Anspruch wird die aktuelle Entwurfsversion des ISO 26262 aus dem Jahr 2016 um Teil 11[3] ergänzt, damit die domänenspezifischen Anforderungen an die Halbleiterentwicklung sichergestellt werden können. Ein domänenübergreifendes Qualitätsverständnis wird Einfluss auf die Entwicklung von künftigen Halbleiterprodukten haben, von denen beide Branchen profitieren.

Im weiteren Verlauf dieses Kapitels wird auf die aktuelle Zulassungssituation von COTS-basierter Avionik eingegangen. Zusätzlich soll dargestellt werden, welche Vorteile die Herstellung einer Vergleichbarkeit der branchenabhängigen Standards in Avionik-Projekten bringen kann.

2.1 Zulassung COTS-basierter Avionik

Einleitend ist zu klären, dass der Begriff *Zulassung* im Zusammenhang mit Avionik-Systemen der Lesbarkeit geschuldet ist. Tatsächlich wird ein Flugsystem von der Zulassungsbehörde als Ganzes genehmigt. Ein Avionik-System wird hierbei nicht einzeln betrachtet, doch muss es konform zu den Regularien entwickelt worden sein damit es *zulassbar* ist. Somit kann im folgenden Avionik-Zulassung als die *Zulassbarkeit* eines Avionik-Systems verstanden werden.

Noch bevor die Hardware von e/e Systemen im Fokus der funktionalen Sicherheit stand, beschrieb man mit dem DO-178[4] die Softwareentwicklung für Luftfahrtanwendungen. Die Absicht war, konkretere Zielvorgaben für die Software-Entwicklung zu definieren, um die abstrakteren Sicherheitsrichtlinien der Zulassungsbehörde zu erfüllen. Mit dem DO-254 [Sp00] wurde 2000 ein Standard veröffentlicht der dieses Ziel für die Hardware-Entwicklung verfolgt. Als übergeordnetes Bindeglied zwischen den DO-178B und dem DO-254 dient die aktuelle Veröffentlichung des Standards ARP4754A [SA10]. Er beschreibt den Entwicklungsprozess für Avionik-Systeme und definiert die beiden Standards als entsprechende Empfehlung für die Hardware und Software-Entwicklung.

[3] Leitfaden zur Anwendung des ISO 26262 für Halbleiter
[4] Aktuelle Veröffentlichung ist der DO-178C aus dem Jahr 2012.

COTS sind Komponenten entwickelt für verschiedene Kunden und unterschiedliche Anwendungen. Der Entwicklungsverlauf bzw. dessen Ergebnis bestimmt alleinig der Hersteller. Die Produkte sind an der Nachfrage am Markt und den akzeptierten Standards ausgerichtet. Der Endkunde hat keinen Einfluss auf den Entstehungsprozess und muss nach dessen Entwicklung nachweisen, ob die Anforderungen für seinen Anwendungsfall erfüllen werden. Mit dieser Tatsache wird in der Luftfahrtindustrie bereits seit Jahrzehnten erfolgreich umgegangen, was durch die im Einsatz befindlichen MPUs in Systemen bewiesen wird [Ye96].

Die heute anvisierten COTS-Komponenten sind komplexer, als die bereits im Einsatz befindlichen MPUs. Die entstandenen Erfahrungen bei der Entwicklung und Zulassung von COTS-basierter Avionik decken die Einschränkungen im praktischen Nutzen des DO-254 auf. Eine umfangreiche Offenlegung von Entwicklungsdaten durch den COTS-Hersteller ist nötig, widerspricht jedoch dem Schutz dieses Wissens. Versuche an dieses Wissen durch *Reengineering* zu gelangen, stellten sich als nicht praktikabel heraus [HB07].

Aus wirtschaftlichen Gründen streben Avionik-Hersteller keine Eigenentwicklung von zentralen Rechenkomponenten wie MPUs und MCUs an und verwenden stattdessen alternative Methoden. Diese sollen nachweisen, dass die COTS-Komponente den Anforderungen der Gesetzgebung und des Avionik-Systems entsprechen. Ein schadhaftes Fehlversagen muss extrem unwahrscheinlich sein. Dabei wird erwartet, dass die Komponente unter allen denkbaren Bedingungen wie beabsichtigt funktioniert und für unerwartete Ereignisse Maßnahmen ergriffen werden.

Aktuelle Ansätze über die Argumentation der Zulassbarkeit von COTS-Komponenten bauen auf einer Kombination folgender Bestandteile auf [Ce14], [Wi15]:

Vorhandene Entwurfsdaten Hat der Hersteller einen strukturierten und anforderungsbasierten Entwicklungsprozess verwendet, so können diese Daten – dessen Einverständnis vorausgesetzt – wiederverwendet werden.

Eigene Erzeugung von Entwurfsdaten Durch Reengineering-Maßnahmen können diese Daten auch nach der Entwicklung erhoben werden.

Betriebserfahrung Die Aussagekraft der Erfahrung kann die Produktreife untermauern, sodass keine systematischen Fehler bei einer ähnlichen Verwendung zum Tragen kommen.

Fehlermaskierung auf Systemebene Das Verhalten bei bestimmten Fehlerszenarien kann nicht in jedem Fall nachgewiesen werden. Durch Anpassung der Systemarchitektur (z.B. strukturelle Redundanz mit unterschiedlichen Komponenten, siehe Kapitel 3) können diese Fehler maskiert werden.

Detaillierte Entwurfsdaten sind notwendig, falls den Informationen zur COTS-Komponente nicht vertraut wird oder deren Aussagekraft nicht ausreicht. Damit die Betriebserfahrung glaubwürdig die Qualität einer Komponente unterstützt, muss diese aufwändig

ermittelt und bewertet werden. Dieses hier beschriebene Vorgehen der COTS-Nachweisführung ist anwendbar für MPUs oder MCUs mit niedriger Komplexität. Ein AMCU zählt nicht hierzu.

2.2 Zielsetzung des Qualitätsvergleichs

In Kapitel 2.1 werden Herausforderungen und aktuelle Lösungsansätze zur Zulassung von Avionik-Systemen beschrieben, wenn eine COTS-Hardware-Komponente eine zentrale Rolle im Systementwurf einnimmt. Die Zulassungsproblematik von COTS-Komponenten ist ein ständiger Wettlauf mit der fortschreitenden Technologie und ohne diese Produkte ein Innovationspotential künftiger sicherheitsrelevanter Avionik gehemmt wäre. Aktuell zulassungskonform entwickelte Komponenten beheben diesen Mangel nicht.

Bisher wurde die Vergleichbarkeit des Qualitätsniveaus zwischen den domänenspezifischen Standrads ISO 26262 Teil 5 und DO-254 nicht untersucht. Ähnliche erbrachte Vergleiche zwischen den Softwarestandards beider Domänen [GHW11], [Le12] geben dazu Anlass, diesen für Hardwarestandards mit einer zielgerichteten Verwertungsperspektive zu erbringen.

Ist das Qualitätsniveau zwischen den Domänen vergleichbar, werden folgende Auswirkungen erwartet:

- Die wiederkehrenden Kosten der Zulassung von COTS-basierter Avionik können reduziert werden. Diese Annahme beruht darauf, dass der Vergleich beider Standards generischer Natur (ohne Projektbezug) ist. So können diese einmal gewonnenen Erkenntnisse für eine Vielzahl von Avionik-Projekten wiederverwendet werden.
- Versuche der Einflussnahme der Luftfahrtindustrie auf die Entwicklung von COTS-Komponenten-Hersteller sind bekannt [CB04]. Ist den Herstellern bewusst, dass die notwendigen prozessspezifischen Anpassungen wirtschaftlich sind, werden sie umgesetzt. Das Risiko der Machbarkeit wird im Vorfeld dieser Forschung erbracht.

3 Sicherheitsbedenken aus dem Avionik-Sektor und Steigerung der Sicherheit

Wie bereits in Kapitel 2.1 beschrieben ist, liefert der DO-254 keine Hilfestellung für die zulassungskonforme Entwicklung von COTS-basierter Avionik. Um den Einsatz von COTS MCUs dennoch zu ermöglichen entstand durch die Federal Aviation Administration (FAA) zu diesem Thema in Zusammenarbeit mit namhaften Firmen aus dem Avionik-Bereich 2011 eine Forschungsarbeit die sich mit dieser Problematik befasst. Aus dem daraus resultierenden „Handbook for the Selection and Evaluation of Microprocessors for Airborne Systems“ [FA11] werden die Sicherheitsanforderungen behandelt die auch auf AMCUs angewendet werden müssen. In diesem Kapitel wird bewertet, ob diese von AMCUs erfüllt werden können. Die Beschreibung eines Redundanzkonzeptes ermöglicht die Erkennung und Toleranz weiterer Fehlerfälle.

3.1 Aktuelle Safety-Bedenken aus dem Avionik-Bereich

Im Rahmen der Untersuchungen seitens der FAA konnten insgesamt drei Hauptpunkte ermittelt werden, bei denen hinsichtlich des Einsatzes von AMCUs im Avionik-Bereich noch Sicherheitsbedenken bestehen.

Sichtbarkeit und Debugbarkeit AMCU-Hersteller erlauben in der Regel keinen Einblick in die internen Strukturen ihrer Produkte, da dieses Wissen als Firmengeheimnis angesehen wird. Dies hat den signifikanten Nachteil, dass sich die Hardware nicht mehr bis ins Detail analysieren lässt, um so exakte Vorhersagen hinsichtlich der Ausführungszeit treffen zu können. Durch die Integration der Systemarchitektur auf einem Chip wird es zudem unmöglich, gezielt zwischen den Komponenten Fehler einzuspeisen. Dadurch lassen sich die Sicherheits-Algorithmen nicht mehr, oder nur mit hohem Aufwand auf der Hardware selbst testen.

Konfigurationsprobleme Da es für einen AMCU-Hersteller nicht wirtschaftlich ist, für jeden Kunden eigene Produkte nach einem exakt vorgegebenen Funktionsumfang zu entwickeln, werden die Produkte für den breiten Markt konzipiert. Dieser beinhaltet einen Durchschnitt an Funktionen die in der entsprechenden Domäne üblicherweise benötigt werden. Da die Funktionen anwendungsspezifisch konfiguriert werden müssen, werden Software-Register zur Aktivierung bzw. Deaktivierung verwendet. Die Konfiguration der einzelnen Komponenten erfolgt dabei ebenfalls über in Software ansteuerbare Register. Bedingt durch Software-Fehler oder atmosphärische Einflüsse wie Single Event Upset (SEU) [5] können sich die Konfigurationen einzelner Register unbeabsichtigt ändern.

Gemeinsam genutzte Ressourcen Während die Anzahl der Komponenten innerhalb einer MCU immer weiter steigt, werden einige Komponenten weiterhin von mehreren Teilnehmern gleichzeitig benutzt. Hierzu zählt unter anderem der Hauptspeicher, der sowohl von Prozessoren als auch Direct Memory Access Controller (DMA-C) bedient wird. Hierbei kommt es, wie in Kapitel 4.1 beschrieben, zwangsweise zu Kollisionen, was starke Auswirkungen auf die Ausführungszeit haben kann.

3.2 Maßnahmen zur Steigerung der Safety auf Architekturebene

Ergänzend werden im Bericht der FAA Lösungsansätze behandelt. Hierbei soll durch Arbitrierungsverfahren für gemeinsam genutzte Ressourcen oder einem sogenannten Frame-Lock Ansatz [FA11] eine Entschärfung der Problematik bei den Ausführungszeiten erreicht werden. Trotz der vorgestellten Möglichkeiten kann durch das Fehlen von Entwurfsdaten nicht sichergestellt werden, dass alle Fehlerfälle abgedeckt sind. Hierzu werden zwingend weitere Maßnahmen auf Systemebene benötigt.

[5] Änderung von Werten/Zuständen innerhalb der MCU oder des Speichers durch geladene Teilchen aus der Atmosphäre.

Redundanz Der klassische Redundanzansatz in Form eines Triplex- [vN56] oder Quadruplex-Systems [AB10] bietet den größten Mehrgewinn bezogen auf die Sicherheit des Gesamtsystems. Hierdurch kann durch die Verschaltung mehrerer gleichartiger AMCUs, wie in Abb. 1 gezeigt, ein Fail-Operational-Verhalten erreicht werden – vorausgesetzt das Redundanznetzwerk selbst wurde entsprechend fehlertolerant ausgelegt.

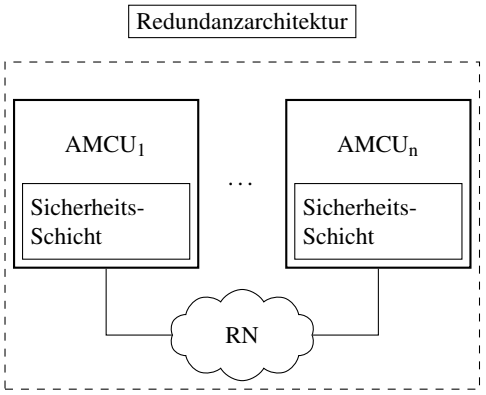


Abb. 1: Redundanzkonzept

Bezüglich der Robustheit gegenüber äußeren Einflüssen gibt es hier Seitens der FAA Vorschriften, welche im DO-160 [Sp10] konkretisiert sind. Für die Realisierung eines solchen Redundanznetzwerkes (in Abb. 1 als RN dargestellt) bietet es sich an, auf die standardmäßig vorhandenen Schnittstellen der verwendeten AMCUs zuzugreifen. Hierzu wurde bereits eine entsprechende Voranalyse in [HH16] durchgeführt. Durch den Aufbau eines solchen Netzwerkes können beispielsweise Konfigurationsprobleme toleriert werden, welche durch äußere Umwelteinflüsse ausgelöst wurden. Auch der komplette Ausfall eines MCU durch Überspannung oder Alterung, kann dadurch toleriert werden. Hierbei bringt es einen entsprechenden Mehrwert, wenn das System um den Aspekt der Hardware-Dissimilarität [Mo99] erweitert wird. Das aktuelle Ausschluss-Kriterium für solche Redundanzarchitekturen sind schlicht die Kosten (siehe [Gr15]), da alle Komponenten innerhalb des Steuergerätes mindestens dreimal vorhanden sein müssen. Andererseits wurden von der Automotive-Industrie bereits erste Bemühungen unternommen um mittels spezieller Duplex-Systeme und Rekonfigurationsmechanismen [Mu15] effizientere Lösungen erreichen zu können.

Dissimilarität Um die Auswirkung von Entwicklungsfehlern der AMCU ausschließen zu können, kann der Redundanzansatz mit verschiedenen MCUs und Intellectual Properties (IPs) verschiedener Hersteller verwendet werden. Der Grad einer hinreichenden Dissimilarität von MCUs muss dabei im Einzelfall geprüft werden.

Der dissimilare Ansatz führt jedoch zu einem signifikanten Anstieg der Entwicklungskosten, da eine Einarbeitung in mehrere MCUs erfolgen muss – zusätzlich zu Entwicklung und Wartung redundanter Software-Versionen für die verschiedenen MCUs.

4 Analyse von Zugriffsstrukturen

In Kapitel 3.2 wird dargestellt, dass einzelne MCUs den Safety-Anforderungen der Luftfahrt nicht genügen, wodurch ein redundanter Betrieb unerlässlich ist. Durch diese Forderung verschärfen sich die in Kapitel 3.1 genannten Bedenken über die Sichtbarkeit und Nutzung gemeinsamer Ressourcen. Die dadurch entstehenden hardwareseitigen Zugriffskonflikte können sich stark auf die Software auswirken, was wiederum zu einer Verletzung von zeitlichen Bedingungen führen kann.

Klassische Avionik-Systeme basieren aktuell auf diskreten CPUs bei denen der E/A-Bus und in vielen Fällen der System- bzw. Speicher-Bus offengelegt ist. Dies bedeutet nicht nur, dass das Busprotokoll (z.B. Arbitrierungsstrategie) zur Analyse offengelegt ist, sondern auch, dass mit externen Messgeräten die Kommunikation verifiziert werden kann.

Durch den Einsatz von hochintegrierten MCUs wird die Analyse des zeitlichen Systemverhaltens, sowie deren Einzelkomponenten und der Anwendungs-Software erheblich erschwert. Um Aussagen über die Zugriffspfade oder auftretender Interferenzen treffen zu können, muss auf integrierte Debug-Schnittstellen oder auf externe Messmethoden zurückgegriffen werden.

4.1 Allgemeine Problematik

In sicherheitskritischen Echtzeitsystemen, wie beispielsweise in Flugsteuerungen der Avionik, werden zyklische Regelschleifen verwendet. Neben dem Sensor-Input und der Aktor-Ansteuerungen werden bei hoch sicherheitskritischen Systemen mit mehrfach redundanter Hardware (vgl. Kapitel 3.2) zudem die Ein- und Ausgaben abgeglichen. In Abb. 2 wird eine typische Verarbeitungskette der Daten dargestellt. Die Ausführung der grau dargestellten Funktionen ist ein synchroner Prozess, für den sich die auftretenden Datenströme gut evaluieren lassen. Die Eingabe der angebundenen Sensoren hingegen läuft asynchron zur Regelschleife ab und steht somit in Konkurrenz zueinander.

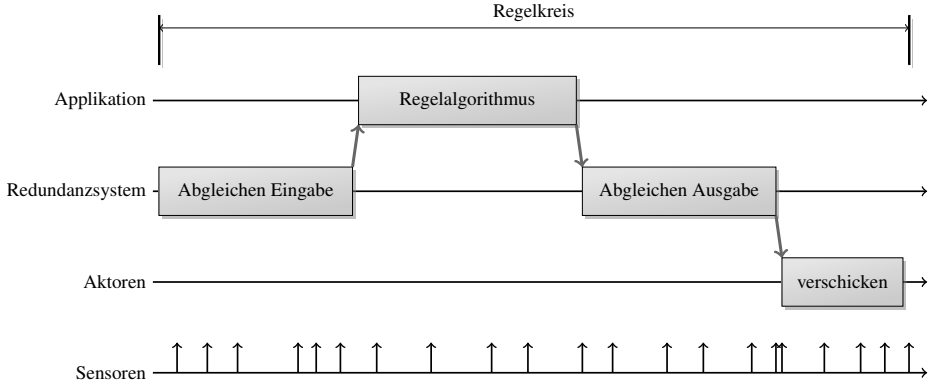
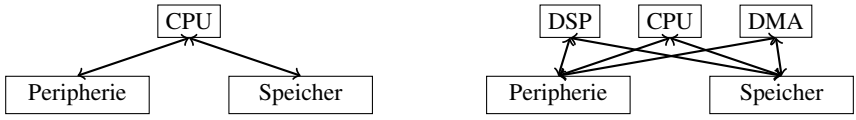


Abb. 2: Erweiterte Regelschleife

Um eine Analyse der Datenströme, insbesondere der Zugriffszeiten auf Register- und Speicherbereiche zu gewährleisten, ist in aktuellen Flugsteuerungsanwendungen das Einlesen der asynchronen Sensordaten ein Teil der (synchronen) Regelschleife. Dies stellt sicher, dass nur eine aktive Komponente (vgl. Abb. 3a) Zugriffe auf die verschiedenen Ressourcen verursacht. Dadurch entstehen keine Konflikte und die Zugriffszeiten auf Adress- und Registerbereiche lassen sich mit einem Maximum (Worst Case) angeben. Ein Nachteil dieser Implementierung ist, dass die Verarbeitung der Datenströme Ressourcen der CPU belegt.



(a) Konkurrenzfreier Zugriff (b) Konkurrierender Zugriff

Abb. 3: Zugriff auf Ressourcen

Diese einfache Datenverarbeitung lässt sich nur mit moderaten Datenraten und wenigen Schnittstellen realisieren. Durch den Anstieg der zu verarbeitenden Daten und abzudeckenden Schnittstellen[6] ist eine effizientere Verarbeitung der Eingabedaten nötig. Dies lässt sich durch den Einsatz spezialisierter Hardware wie DMA-C und digitaler Signalprozessors (Digital Signal Processor, DSPs) realisieren. Mittels eines selbstständigen Zugriffs auf Speicher und Peripherie sowie einer optionalen Datenverarbeitung kann die CPU entlastet werden.

Durch den Einsatz weiterer aktiver Komponenten (vgl. Abb. 3b) lässt sich ein kollisionsfreier Zugriff auf gemeinsame Ressourcen nicht mehr realisieren. Dies führt auch dazu, dass sich die Zugriffszeiten auf Speicher und Register nicht mehr verifizieren lassen. Es kann zwar immer noch eine obere Schranke angegeben werden, wobei diese nur vereinzelt oder in Extremsituationen auftritt und die Vorteile von Speicherdirektzugriff (Direct Memory Access, DMA) und DSP zunichtemacht.

4.2 Abschätzen von Datenströmen

Der pessimistische Ansatz, dass jeder Zugriff eine Kollision verursacht, tritt in modernen MCUs nicht immer auf. Wird von verschiedenen Komponenten auf Ressourcen im System zugegriffen, muss dadurch nicht immer ein Konflikt entstehen. Ein Beispiel hierfür ist die Verwendung eines Crossbar Switch als Verbindungsnetzwerk bei dem nur bei Zugriffen auf identische Zielressourcen Kollisionen auftreten.

[6] Implementierung von Redundanzsystemen mithilfe von on-Chip Schnittstellen (vgl. Kapitel 3.2), Radar- oder Kamera-basierte Fahrerassistenzsystemen (Advanced Driver Assistance Systems, ADAS), etc.

4.2.1 Analyse möglicher Kollisionspfade

Um einen Überblick über die möglichen Kollisionen und deren Eigenschaften zu erhalten lassen sich gezielte Messungen durchführen. Dafür werden einzelne Pfade gegeneinander vermessen und die daraus resultierenden Interferenzen festgehalten.

Als Testaufbau eignet sich ein minimales Programm, welches CPU-getrieben nur einen vorher festgelegten GPIO-Pin zyklisch umschaltet. Hierbei handelt es sich um eine kompakte Befehlsfolge, der im CPU-Cache gehalten wird. Ein Nachladen von Instruktionen aus dem Hauptspeicher ist nicht nötig. Neben den Zugriffen auf den GPIO-Pin werden asynchrone Transfers mithilfe des DMA-C induziert, um Daten beispielsweise von einem UART-Register in den SRAM zu übertragen.

Anhand des Datenbuchs (im aktuellen Beispiel wird ein TI Hercules TMS570LC4357 verwendet, vgl. Reference Manual [Te14]) lässt sich feststellen, dass sowohl das GPIO- als auch das UART-Register einen gemeinsamen Bus verwenden. Treten nun gleichzeitig Transferanfragen auf, so sind hier Kollisionen zu erwarten. Diese lassen sich mit Oszilloskop oder Logic Analyzer aufzeigen, sodass dadurch die daraus resultierende Wartezeit der CPU ausgemessen werden kann. Diese Analyseschritte müssen für jeden möglichen Pfad innerhalb des MCU wiederholt werden.

Neben externen Messmethoden lassen sich mithilfe von Tracing-Schnittstellen Informationen über das System herausführen um die internen Abläufe besser analysieren zu können. Hier kann es jedoch, je nach Implementierung der Tracing-Schnittstelle, zu Problemen kommen. Einige Implementierungen übergeben die Ereignisse, ohne einen internen Zeitstempel zu setzen, an einen Zwischenspeicher. Die gepufferten Nachrichten werden erst durch das Auftreten bestimmter Ereignisse durch den MCU verschickt, wodurch die externen Tracing-Hardware keine exakten Zeitstempel generieren kann. Im Mittel stimmen die interpolierte Ausführungszeiten je Instruktion, jedoch zeigt dies, dass hier der Fokus klar auf den durchschnittlichen Performance-Werten liegt und nicht auf die sicherheitsrelevante Worst Case Execution Time (WCET).

4.2.2 Analyse von Zugriffsmustern

Neben dem Wissen über mögliche Kollisionen sind für Timing-Analysen die tatsächlichen Zugriffsmuster von Applikationen notwendig. Hierunter wird die zeitliche Abfolge von Zugriffen auf bestimmte Adressbereiche verstanden. Ebenfalls können zyklisch wiederkehrende Zugriffsmuster (vgl. Regelschleife, Abb. 2) die Analyse erleichtert, da nur auf einem zeitlich begrenzten Muster Analysen erbracht werden müssen.

Neben den Zugriffsmustern der Applikation lassen sich zudem auch Zugriffsmuster seitens der E/A ermitteln. Können hier keine Muster abgeleitet werden, so lassen sich mit Hilfe der maximalen physikalischen Übertragungsrate einzelner Schnittstellen die Zugriffsraten ermitteln. Es kann davon ausgegangen werden, dass innerhalb der Zeitspanne einer Nachricht maximal eine definierte Anzahl an Unterbrechungen initiiert werden kann.

4.2.3 Abschätzen von Konflikten

Kombiniert man das Wissen aus der Analyse der Zugriffsmuster der Software und aus denen der Schnittstellen, so lassen sich die theoretische Anzahl an maximalen Konflikten errechnen. Dies reduziert die Überschätzung der WCET erheblich, da nur die maximal auftretenden Konflikte berechnet werden und nicht für jede Instruktion von einer Vielzahl von Kollisionen ausgegangen werden muss.

5 Schluss

Bei der Entwicklung von funktional sicheren Systemen wechselten sich die Luftfahrt- und Automobilindustrie, ähnlich eines Staffellaufs, chronologisch bei der Weiterentwicklung von Technologien und Methoden ab. So wie vor der Jahrtausendwende Fly-By-Wire-Systeme entstanden sind, die keiner mechanischen Absicherung bedürfen [Ye96], so hat die Automobilindustrie durch ihre Marktmacht den Sicherheitsgedanken bis zum Hardware-Komponenten-Hersteller, die zu AMCUs geführt haben, transportieren können.

Dieser Forschungsbeitrag versteht sich als Fortsetzung dieses Staffellaufes. Es werden Untersuchungen beschrieben die u. a. notwendig sind um AMCUs in sicherheitskritischen Avionik-Systemen, mit dem Anspruch der Zulassbarkeit, einsetzen zu können. Indem die Luftfahrtindustrie den Stab weiterträgt, werden ebenso neue Impulse für den Automobilmarkt gesetzt. Diese können einen Mehrwert für neue sicherheitsrelevante Automotive-Systeme im Bereich des autonomen Fahrens darstellen.

Danksagung

Diese Veröffentlichung wird unterstützt durch:

- das Projekt FORMUS^{3}IC “Multi-Core safe and software-intensive Systems Improvement Community”, Förderkennzeichen AZ-1165-15, der Bayerische Forschungsstiftung,
- das Open Innovation for RPAS (OPIRA) Projekt, finanziert durch das Luftfahrtforschungsprogramm V (LuFo V5-1), Förderlinie “Technologie” des Bundesministeriums für Wirtschaft und Energie und
- der von Airbus Defense and Space finanzierten Stiftungsprofessur “Systemtechnik für sicherheitskritische Software”, unterstützt durch den “Stifterverband für die Deutsche Wissenschaft e.V.”

Literaturverzeichnis

[AB10] Audsley, N. C.; Burke, M.: Distributed Fault-Tolerant Avionic Systems – A Real-Time Perspective. 2010.

[CB04] Cole, P.; Beeb, M.: Safe COTS graphics solutions: impact of DO-254 on the use of COTS graphics devices for avionics. In: The 23rd Digital Avionics Systems Conference (IEEE Cat. No.04CH37576). Institute of Electrical and Electronics Engineers (IEEE), 2004.

[Ce14] Certification Authorities Software Team: Compliance to RTCA DO-254/ EUROCAE ED-80, "Design Assurance Guidance for Airborne Electronic Hardware", for COTS Intellectual Properties Used in Programmable Logic Devices. Bericht 33, Federal Aviation Administration, August 2014.

[FA11] FAA: Handbook for the Selection and Evaluation of Microprocessors for Airborne Systems. Bericht, 2011.

[FK06] Forsberg, Hakan; Karlsson, Kristoffer: COTS CPU Selection Guidelines for Safety-Critical Applications. In: 25TH Digital Avionics Systems Conference. Institute of Electrical and Electronics Engineers (IEEE), oct 2006.

[GHW11] Gerlach, Matthias; Hilbrich, Robert; Weißleder, Stephan: Can cars fly? from avionics to automotive: Comparability of domain specific safety standards. In: Proceedings of the Embedded World Conference. 2011.

[Gr15] Grave, Rudolf: Autonomous Driving – From Fail-Safe to Fail-Operational Systems. TechDay December2015, 2015.

[HB07] Hilderman, Vance; Baghi, Tony: Avionics certification: A complete guide to DO-178 (software), DO-254 (hardware). Avionics Communications, Leesburg, VA, 2007.

[HH16] Hiergeist, Sebastian; Holzapfel, Florian: Fault-tolerant FCC Architecture for future UAV systems based on COTS SoC. 2016.

[Le12] Ledinot, Emmanuel; Gassino, Jean; Blanquart, Jean-Paul; Boulanger, Jean-Louis; Quéré, Philippe; Ricque, Bertrand: A cross-domain comparison of software development assurance standards. ERTS, 2012.

[Mo99] Montenegro, Sergio: Sichere und fehlertolerante Steuerungen: Entwicklung sicherheitsrelevanter Systeme. Hanser, München [u.a.], 1999.

[Mu15] Much, Alexander: The Safe State: Design Patterns and Degradation Mechanisms for Fail-Operational Systems. safetronic.2015, 2015.

[SA10] SAE Aerospace: Guidelines for Development of Civil Aircraft and Systems. Bericht ARP4754, SAE International, Dezember 2010.

[Sp82] Special C. of RTCA: DO-178: Software Considerations in Airborne Systems and Equipment Certification. RTCA, Dezember 1982.

[Sp00] Special C. of RTCA: DO-254, Design Assurance Guidance for Airborne Electronic Hardware. RTCA, April 2000.

[Sp10] Special C. of RTCA: DO-160G: Environmental Conditions and Test Procedures for Airborne Equipment. RTCA, 2010.

[Te09] Technical Committee 22: ISO/DIS 26262 - Road vehicles – Functional safety. Bericht, International Organization for Standardization, Geneva, Switzerland, Juli 2009.

[Te14] Texas Instruments: . TMS570LC43x 16/32-Bit RISC Flash Microcontroller. Texas Instruments, Mai 2014.

[vN56] von Neumann, J.: Probabilistic Logics and Synthesis of Reliable Organisms from Unreliable Components. 1956.

[Vo13] Volpe National Transportation Systems Center: Unmanned Aircraft System (UAS) Service Demand 2015-2035: Literature Review and Projections of Future Usage. Bericht DOT-VNTSC-DoD-13-01, U.S. Department of Transportation, 2013.

[Wi15] Wilkinson, Chris: Obsolescence and Life Cycle Management for Avionics. Bericht, Federal Aviation Administration, November 2015.

[Ye96] Yeh, Y.C.: Triple-triple redundant 777 primary flight computer. In: IEEE Aerospace Applications Conference. Proceedings. Institute of Electrical and Electronics Engineers (IEEE), 1996.