

Untersuchung zum Einsatz von ARM Cortex-M Prozessorkernen in voll digitalisierten Wechselrichtern



Verfasser: Prof. Dr.-Ing. Bernhard Frenzel
Mechatronische Systeme
Fakultät Maschinenbau/Umwelttechnik
Ostbayerische Technische Hochschule Amberg
Kaiser-Wilhelm-Ring 23
92224 Amberg
b.frenzel@oth-aw.de

Datum: 17. März 2014

Inhalt

1	ABSTRACT	3
2	EINFÜHRUNG	4
3	ZIEL DER UNTERSUCHUNG	5
4	HARDWARE	6
5	ECHTZEITBEDINGUNGEN.....	9
6	SOFTWARE	11
7	ALGORITHMEN.....	12
8	ERGEBNIS.....	14
9	ABKÜRZUNGEN	15
10	LITERATUR.....	15

1 Abstract

Zum Betrieb moderner elektrischer Antriebe und zur Ankopplung von Solarmodulen an unsere elektrischen Energieversorgungsnetze werden in großer Stückzahl leistungselektronische Stellglieder benötigt, die Gleichspannung in Wechselspannung umwandeln. Diese Wechselrichter sind in heutiger Zeit nahezu ausnahmslos digitalisiert. Eine zentrale Rolle spielen dabei die verbauten Prozessoren, die preisgünstig, leistungsfähig und leicht portierbar sein müssen. Besonderes Augenmerk liegt dabei auf Prozessorarchitekturen, die nicht an einen speziellen Hersteller gebunden sind. Dies ermöglicht einen vergleichsweise schnellen und damit preisgünstigeren Wechsel zu einem anderen Hersteller, wenn Prozessoren abgekündigt werden. Eine solche Prozessorarchitektur bietet die Firma ARM (Advanced Risc Machines Ltd.) mit dem Cortex-M Kern speziell für den Verbau in eingebettete Systeme an, der weltweit inzwischen eine außerordentliche Verbreitung gefunden hat. Die vorliegende Untersuchung überprüft daher Prozessoren mit ARM Cortex-M Architektur hinsichtlich ihrer Verwendbarkeit in voll digitalisierten Wechselrichtern. Nach einer Gegenüberstellung in Bezug Preis und Ausstattung verschiedener Prozessortypen fiel die Wahl auf eine Variante des STM32F103 der Firma ST Microelectronics. Zur Verifikation wurde auf Basis eines Evaluation Boards dieses Prozessors ein Echtzeit-Scheduler mit Hintergrundmonitor zur Abarbeitung der Wechselrichtersoftware entwickelt. Anhand des Modellsystems konnten Algorithmen aus anderen Projekten zum Betrieb in voll digitalisierten Wechselrichtern erfolgreich implementiert werden. Es zeigte sich, dass die Cortex-M Kerne im Vergleich zu bisher im Labor Mechatronik verwendeten Prozessoren bei der Hälfte des Preises mehr als das doppelte an Rechenleistung zur Verfügung stellen. Als Ergebnis ist die ARM Cortex-M Architektur somit voll umfänglich geeignet, um leistungsstarke und preisgünstige Wechselrichtersysteme aufzubauen.

For the operation of modern electric drives and for coupling solar modules to our electrical power grids, a large number of power electronic devices converting DC to AC voltage are required. Nowadays, these inverters are digitized almost without exception. For this, the installed processors play a prominent role, which shall be low-priced, powerful and portable. Particular attention is given to processor architectures that are not dependent on a single supplier. In case of a processor's obsolescence, this allows a comparably fast and therefore cheaper change to another manufacturer. Such a processor architecture is provided by the company ARM (Advanced RISC Machines Ltd.), known as Cortex-M. This architecture was specifically designed for installation in embedded systems. Up until now, this type of processor has widely disseminated throughout the world. Therefore, the present study examined processors with ARM Cortex-M architecture in terms of usability for fully digitized inverters. A comparison of price and features for different types of processors revealed the STM32F103 from ST Microelectronics to be a promising candidate. On basis of an evaluation board a real-time scheduler with background monitor was developed. For verification, different algorithms already used in other fully digitized inverter projects were successfully implemented. The study demonstrated that the Cortex-M cores have more than double the processing power at half the price compared to previously used processors. As a result, the ARM Cortex-M architecture is fully suitable to build high-performance and low-cost inverter systems.

2 Einführung

In der heutigen Zeit sind die überwiegende Mehrzahl der am Markt erhältlichen Wechselrichter voll digitalisiert. Diese Baugruppen werden flächendeckend sowohl im Bereich der regenerativen Energiegewinnung zum Beispiel in Kombination mit Solarmodulen in der Photovoltaik, als auch zur gezielten Veränderung der Drehzahl, der Lage oder des Drehmoments in Verbindung mit elektrischen Maschinen und Antrieben verwendet.

Wechselrichter dienen zur Erzeugung sinus- oder kosinusförmiger Spannungs- oder Stromverläufe größerer elektrischer Leistungen und werden von Gleichstrom- oder Gleichspannungsquellen gespeist. Die Amplitude, Frequenz und der Nullphasenwinkel der Ausgangsspannungen oder –ströme sind einstellbar und an die jeweilige Anforderung anzupassen. Die meisten Wechselrichter neuerer Bauart sind Gleichspannungs-Zwischenkreis-Wechselrichter, die aus einer Gleichspannungsquelle –bei mobilen Anwendungen häufig aus Batterien [1]– gespeist werden und eine oder mehrere sinusförmige Ausgangsspannungen erzeugen. Aus Gründen der Minimierung der Verluste wird dabei die Sinusform der Ausgangsspannung nur im Mittel eingestellt. Durch Veränderung des Puls/Pause Verhältnisses wird also der Mittelwert der Ausgangsspannung verändert, grundsätzlich wird jedoch immer die gesamte Gleichspannung auf den Ausgang zugeschaltet oder von diesem getrennt.

Dieses als Pulsweitenmodulation (PWM) bekannte Verfahren wird heute in fast allen Wechselrichtern mit Mikroprozessorsystemen realisiert. Bei Verwendung in Kombination mit elektrischen Antrieben wird mit diesem Verfahren im Mittel die Spannung an den Motorwicklungen der Drehfeldmaschinen –Asynchron- oder Synchronmaschine– in Abhängigkeit von den gerade gemessenen Motorströmen bzw. der Rotorlage eingestellt.

Ein geeignetes Prozessorsystem muss daher in möglichst hoher Auflösung zusammen mit kurzen Zykluszeiten Algorithmen bearbeiten, das in Abhängigkeit von den Sollwerten und den aktuellen Messwerten die bei Drehstrommaschinen notwendigen drei zunächst unabhängigen sinusförmigen Spannungen für die Motorwicklungen zur Verfügung stellt. Die Bearbeitung in zeitlich kurzen Zyklen –typisch sind etwa 20 kHz bzw. 50 μ s– bedeutet zwangsweise auch die Forderung einer hohen Rechenleistung. Zusätzlich muss ein Prozessorsystem eines Wechselrichters Kontrollfunktionen übernehmen, um zu gewährleisten, dass der Betriebszustand des Antriebs oder der Anlage jederzeit im zulässigen Arbeitsbereich liegt. Ebenso muss ein Monitorprogramm zur Verfügung gestellt werden, das dem Benutzer den aktuellen Systemzustand anzeigt oder eine Parametrierung des Wechselrichters erlaubt, ohne jedoch die Algorithmik innerhalb eines Zyklus zu stören oder gar zu unterbrechen. Auch muss fast immer für höhere Zwischenkreisspannungen eine galvanische Isolation und eine Datenbusanbindung zu einer übergeordneten Steuerung vorhanden sein, um nur ein paar Anforderungen zu nennen.

Für ein solches Wechselrichtersystem wird daher ein schneller, hochauflösender und flexibler Prozessor mit einer Software mit Echtzeitfunktionalität und vielfältiger Hardware benötigt. Zudem ist ein äußerst wichtiger Gesichtspunkt bei der Prozessorwahl, inwieweit die einmal entwickelte Hardware zu konkurrenzfähigen Preisen angeboten werden kann und wie leicht die erstellte Software z.B. aufgrund von Obsoleszenzen auf Prozessoren anderer Hersteller portiert werden kann. Gerade ein Prozessorwechsel als zentraler Baustein einer Wechselrichterhardware kann dabei eine Hürde sein, vor der Entwicklungsabteilungen zunächst zurückschrecken, da die Einarbeitung in einen neuen Prozessor kosten- und zeitintensiv ist.

Ein möglicher Ausweg aus dieser Problematik ist, auf Prozessorkerne zu setzen, die von verschiedenen Herstellern angeboten werden, jedoch herstellerunabhängig immer dieselbe

Architektur besitzen. Prozessorkerne mit der sog. ARM-Architektur der Firma ARM (Advanced RISC Machines Ltd.) bieten diese Eigenschaft. Die Firma produziert selbst keine Prozessoren, sondern spezifiziert lediglich deren Architektur samt passendem Befehlssatz. Diese wird anschließend den Halbleiterherstellern zur Implementierung in Lizenz zur Verfügung gestellt. Die Hersteller ergänzen die von ARM spezifizierten Kerne durch eigene, zusätzliche Peripherie, der Kern samt Interrupt Controller ist jedoch bei allen Prozessoren gleich. So ist es nicht weiter verwunderlich, dass bereits im Jahr 2010 mehr als zwei Milliarden ARM Prozessoren pro Jahr ausgeliefert wurden [2].

Alle heute führenden Halbleiterhersteller weltweit sind Lizenznehmer von ARM Prozessorkernen. Je nach Anwendung werden verschiedene Versionen der ARM-Architektur realisiert. Für den Bereich Mobile Computing für Tablets oder Smartphones (z.B. Apple iPad Air, Samsung S4, Apple iPhone 5S: ARM Cortex A Serie) werden andere Prozessorarchitekturen angeboten als für den Embedded Bereich. Prozessoren für diese Anwendungen benötigen einen weniger komplexen Befehlssatz und eignen sich daher auch für den Einsatz in Wechselrichtern.

Speziell für den Embedded Bereich wurde die Cortex-M-Serie entworfen, in der die erstmals 2004 und 2010 vorgestellten Versionen Cortex-M3 und Cortex-M4 sich besonders zum Einsatz als Prozessoren in Wechselrichtern anbieten. Die Version M4 ist ein im Wesentlichen um Signalprozessorbefehle erweiterte M3 Version mit höherer Rechenleistung. Optional ist auch eine Gleitkommaeinheit verfügbar. Für preisgünstigere Lösungen wird man jedoch auf die Version M3 zurückgreifen. Für diese Version ist aufgrund der Popularität und des guten Preis-/Leistungsverhältnisses zwischenzeitlich eine Vielzahl an Prozessorvarianten erhältlich. Beim Hersteller ST Microelectronics beispielsweise kann der Kunde alleine nur für die M3 Version aus über 180 verschiedenen Prozessorvarianten auswählen [3]. Bei Rechenleistungen von bis zu 150 DMIPS sind diese 32-bit Prozessoren preislich im Bereich von 8-bit Prozessoren angesiedelt, jedoch um ein Vielfaches leistungstärker als letztere.

3 Ziel der Untersuchung

Ziel der Studie war es, aufgrund der in der Einleitung genannten Vorteile einen Prozessor mit ARM Cortex-M3 Kern hinsichtlich Verwendbarkeit zum Einsatz in voll digitalisierten Wechselrichtern näher zu untersuchen und ggf. dessen Verwendbarkeit für den genannten Einsatzfall nachzuweisen.

Um möglichst preisgünstige Prozessorvarianten verwenden und um möglichst kleine Bauräume der Wechselrichter-Hardware auch mit Hinblick auf mobile Anwendungen realisieren zu können, sollte dabei kein umfangreiches Echtzeitbetriebssystem eingesetzt werden, das ggf. externe Speicher benötigen würde und dessen Funktionalität im Detail schwer zu überprüfen wäre.

Die Software sollte so beschaffen sein, dass sie möglichst klar strukturiert, übersichtlich und einfach zu validieren ist und in preisgünstige Prozessorvarianten mit selbst geringer Speicherdichte von nur einigen kBytes zu integrieren ist. Auch sollte gezeigt werden, dass der Thumb Befehlssatz des Cortex M3 Kerns bestens zum Einsatz in voll digitalisierten Wechselrichtern geeignet ist, obwohl es sich dabei um einen reduzierten ARM Befehlssatz handelt, jedoch mit dem Vorteil größerer Speicherdichte [4-6].

Zum Leistungsnachweis sollte daher ein vollständig in Assembler entworfener Echtzeit-Scheduler auf einem Modellwechselrichter implementiert werden, mit dem die Algorithmen simultan von einem Antrieb bis zu maximal vier Antrieben gerechnet werden können.

4 Hardware

4.1. Prozessor

Als Hardware wurde ein ARM Cortex-M3 Prozessor ausgewählt, der mindestens vergleichbare Leistungsmerkmale wie die in früheren Projekten eingesetzten 32-bit Prozessoren für voll digitalisierte Wechselrichtersysteme hat, jedoch preislich deutlich unter den bisher verwendeten Prozessoren liegen sollte. Ein Leistungs- und Preisvergleich zweier ausgewählter ARM Cortex Prozessoren der Versionen M3 und M4 mit bisher im Labor Mechatronik verwendeten Prozessoren für Wechselrichtersysteme zeigt Abbildung 1.

Performance Comparison of Different CPUs – for embedded inverter applications –									
CPU	Manufact.	Core		Clock	DMIPS	Algorithm	CPU Time	Algorithm	CPU Time
MC68332	Freescale	68k	32-bit	20 MHz	4	Sinus	3.40 μ s	Alg_res with atan2	23.8 μ s
DSP56F803	Freescale	DSC	16-bit	80 MHz	40		0.65 μ s		3.4 μ s
STM32F103	ST Micro	Cortex-M3 32-bit		72 MHz	90		0.47 μ s		2.0 μ s
STM32F407	ST Micro	Cortex-M4 32-bit		168 MHz	210		0.20 μ s ^(*)		0.86 μ s ^(*)

Chip	Pins	Flash/kB	RAM/kB	Price p.u. ⁽¹⁾	Price p.u. ⁽²⁾
MC68332GCEH20	132	—	2	14.85 €	22.62 €
DSP56F803BU80E	100	64	4	9.27 €	7.87 €
STM32F103VCT6	100	256	48	4.72 €	3.87 €
STM32F407VGT6	100	1024	192	7.17 €	7.74 €

- Test algorithms with table lookup for trigonometric functions (sin, atan2)
- MC68332 features an assembly lookup instruction (tbls.w)

^(*) Time scaled using STM32F407 to STM32F103 DMIPS ratio

⁽¹⁾ Farnell net price without discount for >300 pcs, (November 28, 2013)

⁽²⁾ DigiKey net price without discount for 1000 pcs, (November 28, 2013)

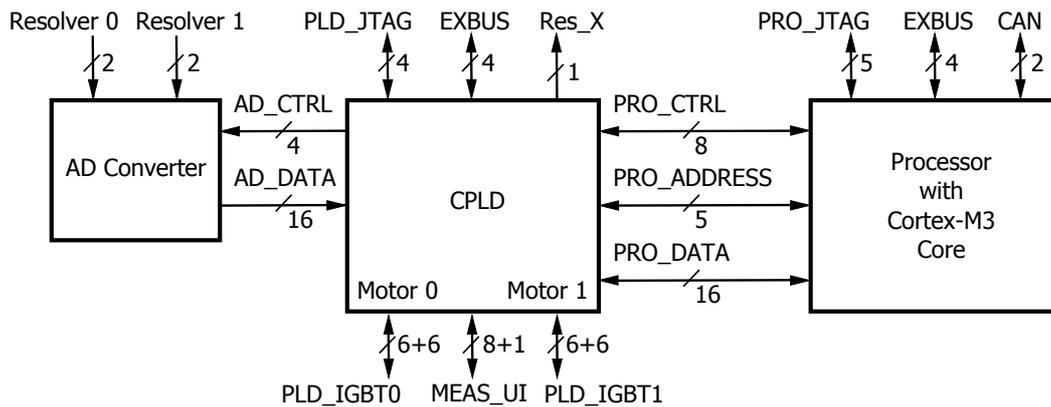
Abbildung 1: Leistungs- und Preisvergleich verschiedener Prozessoren für voll digitalisierte Wechselrichter

Beide dargestellten Cortex-M Prozessorvarianten verfügen neben den im Kern bereits standardmäßig integrierten Komponenten zusätzlich über alle Baugruppen, die der Modellwechselrichter benötigt, wie etwa mindestens einen eingebauten CAN-Controller und eine USART-Schnittstelle für die Monitoring-Funktion [5].

Vorabuntersuchungen ergaben, dass die Rechenzeit des nach Abbildung 1 deutlich günstigeren Cortex-M3 Prozessors für den Modelwechselrichter ausreichend ist. Daher wurde als Prozessor der Prozessor STM32F103VCT6 [7] der Firma ST Microelectronics gewählt.

4.2. Timing Master

Der Vorteil des gewählten Prozessortyps ist, dass er bereits über eine externe Datenschnittstelle (FSMC) zur Anbindung weiterer Prozessoren, Speicher oder programmierbarer Logikbausteine verfügt. Grundsätzlich benötigt ein Prozessor in einem Wechselrichtersystem eine solche Datenschnittstelle zwar nicht. Aus Sicherheitsgründen ist es dennoch häufig vorteilhaft, eine externe, vom Prozessor unabhängige Hardware zu haben, die als Timing-Master fungiert und die Zeitscheibe des Prozessor durch einen einzigen, im System vorkommenden Interrupt zyklisch weiterschaltet (siehe 5 Echtzeitbedingungen). Diese Hardware übernimmt im Modellwechselrichter auch aus Synchronisationsgründen die Pulsweitenmodulation zur Ansteuerung der Motoren, die Steuerung des externen AD-Wandlers zur Lageerfassung der Motoren, die Resolverversorgung sowie die Aufbereitung der Strom- und Spannungsmesssignal(e). Eine Übersicht eines solchen Systems für zwei Motoren zeigt Abbildung 2.



Function	Type	Remarks
AD Converter	AD7655ASTZ	Simultaneous sampling, 16 bit resolution
CPLD	Altera MAX II Series	Atmel MACH XO2 Series for cost sensitive applications
Processor	STM32F103VCT6	Cortex-M3, 72 MHz, 90 DMIPS, CAN, USART, 256 kB Flash, 48 kB RAM

Abbildung 2: Systemübersicht eines Modellwechselrichters für 2 Motoren mit CPLD als Timing Master

Um möglichst flexibel verschiedene Verfahren implementieren zu können, wurde als externe und vom Prozessor unabhängige Hardware ein bereits im Labor Mechatronik der Technischen Hochschule Amberg-Weiden vorhandener, programmierbarer Logikbaustein (CPLD) verwendet, der über eine 16-bit breite Datenleitung an den FSMC des ARM-Prozessors angeschlossen wurde. Die Hardware im Baustein wurde mit der Hardware Beschreibungssprache (HDL) Verilog realisiert, deren Syntax mit der weit verbreiteten Programmiersprache C verwandt ist.

Eine parallele Anbindung wurde aufgrund der hohen Abtastraten einer seriellen Anbindung vorgezogen, um in möglichst kurzer Zeit die bei vier Motoren maximal 4x3 PWM Kommandos an das CPLD zu übertragen. Die Darstellung nach Abbildung 3 zeigt einen mit dem Logic Analyzer Tektronix TLA6202 [8] aufgezeichneten Datentransfer für insgesamt 2 Motoren.

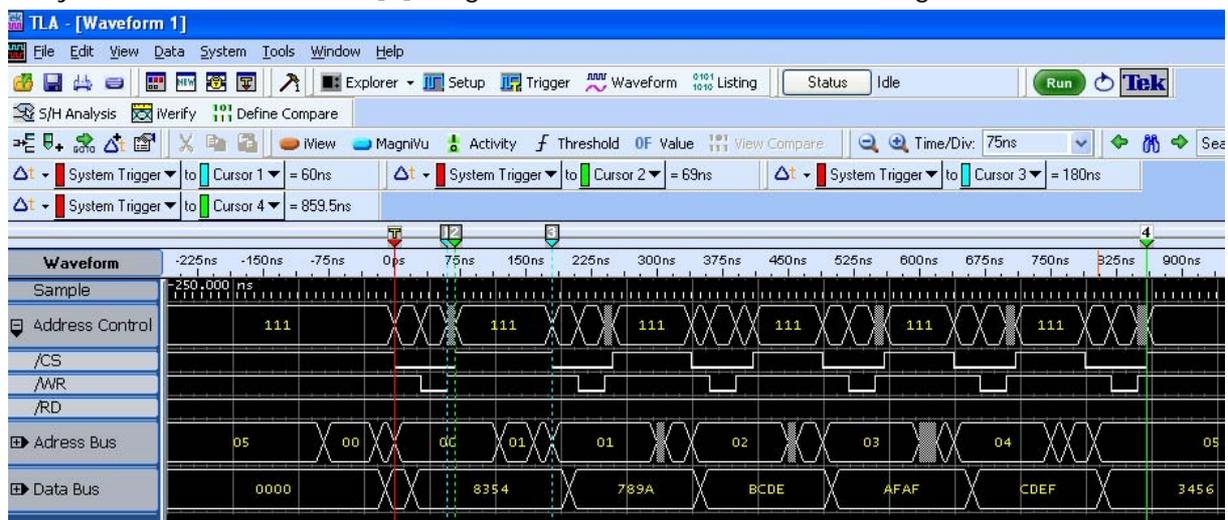


Abbildung 3: Datentransfer (PWM-Kommandos) von ARM Cortex-M3 Prozessor zum CPLD bei zwei Motoren

In der Darstellung werden die 6x16-bit breiten PWM-Kommandos zur Ansteuerung der Halbbrücken des Motors 0 (Adressen 0x00...0x02) und des Motors 1 (Adressen 0x03...0x05) innerhalb von etwa 860 ns übertragen.

4.3. Rotorlageerfassung

Als Rotorlagegeber werden die besonders bei elektrischen Antrieben häufig anzutreffenden Resolver verwendet, da diese Geber besonders robust sind und –verglichen mit Encodern und anderen Drehgebern– bei rauen Umgebungsbedingungen wie etwa bei hohen Temperaturen geringe Fehlerraten aufweisen [9].

Zur Lageerfassung der Rotorstellungen der Motoren werden in Abbildung 2 nicht die bereits in den Prozessoren integrierten AD-Wandler verwendet, da deren 12-bit Auflösung zu gering ist. Ursächlich dafür ist, dass der über den 4-Quadranten-Arkustangens zu berechnende Rotorwinkel zur Gewinnung von Drehzahl bzw. Drehbeschleunigung noch bis zu zweimal zu differenzieren ist. Dadurch verschlechtert sich das SNR erheblich und eine geeignete Regelung wäre nur noch mit unzureichender Regelgüte realisierbar. Zudem kann eine Wandlung nur in den Pulspausen der PWM-Muster erfolgen und muss mit der Pulsung synchronisiert werden (siehe 5 Echtzeitbedingungen).

4.4. Strom- und Spannungsmessung

Die Strommessung erfolgt bei Wechselrichtersystemen sehr häufig durch Mittelwertbildung der Messsignale über eine Abtastperiode. Dadurch bleibt die Strommessung durch kurze Störungen aufgrund der Pulsung der Leistungsstufe unbeeinflusst. Besonders einfach und mit minimalen Hardwareaufwand lässt sich dies dadurch realisieren, dass man Strommesswiderstände in Kombination mit einem bereits galvanisch isolierten $\Sigma\Delta$ -Modulator einsetzt [10]. Man kann dann direkt und ohne Decimation durch Summation des $\Sigma\Delta$ -Signals des Modulators über einen Zähler im CPLD die Mittelwertbildung vornehmen. Bei einer Periodendauer von beispielsweise 50 μs und einer Taktfrequenz des Modulators von 20 MHz folgt eine Auflösung von 10-bit. Nimmt man exemplarisch Strangströme von bis zu $\pm 200\text{ A}$ an, entspricht dies einer ausreichenden Auflösung des Stromes von $I_{\text{LSB}} \approx 400\text{ mA}$. Eine im Labor Mechatronik der Hochschule Amberg entwickelte, bis 2500 V Wechselspannung vollständig isolierte $\pm 200\text{ A}$ Messeinheit sowie deren Messfehler zeigt Abbildung 4. In die Strommesseinheit wurde zusätzlich eine isolierte Versorgung integriert [11]. Somit werden nur nicht isolierte Versorgungsspannungen zum Betrieb benötigt.

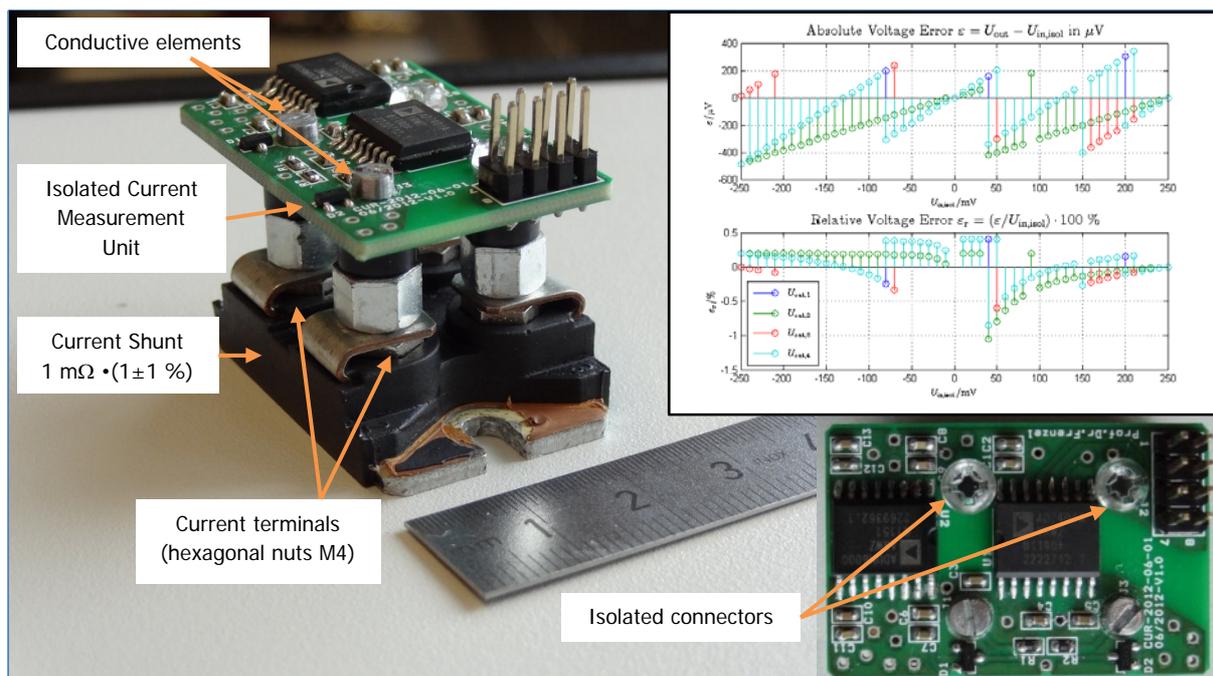


Abbildung 4: Strommesseinheit bis 200 A Dauerstrom, 2,5 kV Isolationsspannung sowie ermittelter Messfehler, gemessen an vier Mustern mit Sourcemeter Keithley 2410 im Bereich $\pm 250\text{ mV}$ Shuntspannung

Für Ströme über 200 A Dauerstrom wird man eine Erfassung der Strangströme mit HALL-Sensoren vorziehen, da die Verluste der Shuntmessung zu hoch werden.

Die isolierte Strommeseinheit lässt sich problemfrei auch zur Messung der Zwischenkreisspannung des Wechselrichters verwenden. Hierzu wird lediglich die isolierte Strommeseinheit an einen hochohmigen Spannungsteiler anstelle des Strommesswiderstands angeschlossen.

4.5. Debugging und Monitoring

ARM Cortex Prozessoren haben im Vergleich zu älteren ARM Modellen eine völlig neu überarbeitete Debug-Schnittstelle, die eine Erweiterung einer JTAG-Schnittstelle hinsichtlich real-time Funktionalität darstellt [2]. Dieses sog. CoreSight Debug System ist über einen internen Bus an den Kern angeschlossen und nicht –wie bei anderen ARM Kernen- in diesen integriert. Dadurch ist ein on-chip Debugging selbst im Niedrigenergie- oder sogar Schlafmodus des Prozessors möglich. Auch kann der Debugger auf Register und Speicherinhalte im laufenden Betrieb zugreifen [6]. Die zusätzlich benötigte Debugging Interface Hardware ist dadurch vergleichsweise einfach gestaltet.

Als Debugging Interface Hardware wurde für den Modellwechselrichter der kostengünstige USB ST-LINK/V2 [12] verwendet, der sowohl eine Übertragung des Programms in den FLASH Speicher als auch das on-chip Debugging via JTAG erlaubt. Die Monitoring-Schnittstelle war PC-seitig eine RS232-Schnittstelle (COM), die bei modernen Rechnern häufig nicht mehr on-board zur Verfügung gestellt wird, aber mit einer preisgünstigen PCI Einsteckkarte realisiert werden kann, siehe z.B. [13].

5 Echtzeitbedingungen

Beim Aufbau voll digitalisierter Wechselrichter ist grundsätzlich das Timing von ganz entscheidender Bedeutung für den störungsfreien Betrieb und für ein deterministisches Schaltverhalten. Grund sind die signifikanten elektromagnetischen Störungen bei Schaltvorgängen an Zwischenkreisspannungen von mehreren hundert Volt bzw. mit induktiven Strömen von mehreren hundert Ampere. Analoge und zum Betrieb eines Wechselrichters notwendige Messvorgänge können meist nur in den Schaltphasen der Leistungshalbleiter durchgeführt werden. Außerdem ist eine simulative Vorabuntersuchung bei Einsatz einer deterministischen Zeitscheibensteuerung unter Echtzeitbedingungen wesentlich einfacher oder gar erst möglich.

Grundsätzlich verfügt der ARM Cortex-M3 Kern über einen 24-bit Timer (SYSTICK Timer), der sich für die Verwendung einer Zeitscheibensteuerung anbietet. Treten jedoch Laufzeitfehler während des Betriebs auf, besteht die Gefahr von Fehlschaltungen der vergleichsweise teuren Leistungshalbleiter des Wechselrichters, was zu deren Zerstörung führen kann. Daher wird als Timing Master meist eine externe Schaltung in Form eines programmierbaren Logikbausteines gewählt, der im zeitlichen Raster der Zeitscheibensteuerung eine im ARM Prozessorsystem vorkommende Unterbrechung (IRQ) einleitet und damit einen Zeitscheibenwechsel auslöst. Wird innerhalb eines definierten Zeitintervalls die Unterbrechung nicht quittiert und neue Schaltsignale an den Pulsbreitenmodulator des programmierbaren Logikbausteins verschickt, schaltet dieser die Endstufen selbstständig auf Null Volt Ausgangsspannung.

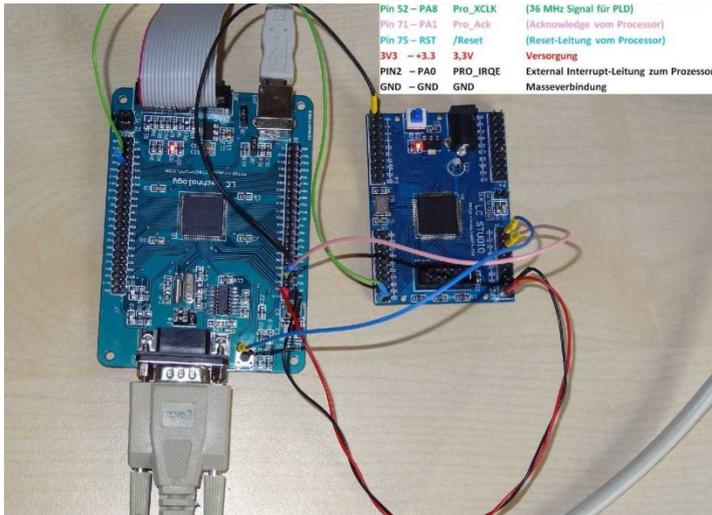


Abbildung 5: Minimal notwendige Verbindungen zwischen ARM Cortex-M3 Prozessor (links) und CPLD (rechts) für einen Unterbrechungs-/Quittungsmechanismus zur Zeitscheibensteuerung



Abbildung 5: Zeit für Context Switch vom IRQ bis zum Vordergrund

Ein solcher Unterbrechungs-/Quittungsmechanismus ist nach Abbildung 5 vergleichsweise einfach zu realisieren und bietet ein hohes Maß an Sicherheit vor Zerstörung der Leistungshalbleiter. Ebenso einfach lassen sich Rechenzeitüberschreitungen detektieren, da eine Unterbrechung nur dann auftreten darf, wenn die Vordergrundpartition vollständig abgearbeitet wurde und sich der ARM Prozessor in einer Hintergrundpartition befindet. Da der Interrupt-Controller des Cortex-M allerdings keine Unterbrechung einer ISR durch einen Interrupt gleicher Priorität zulässt, löst im Scheduler der externe IRQ eine Unterbrechung geringerer Priorität aus und beendet sich anschließend selbst. Damit ist sichergestellt, dass ein neuerlicher, externer IRQ die niederpriorie ISR, in der die Vordergrundpartition läuft, unterbrechen kann.

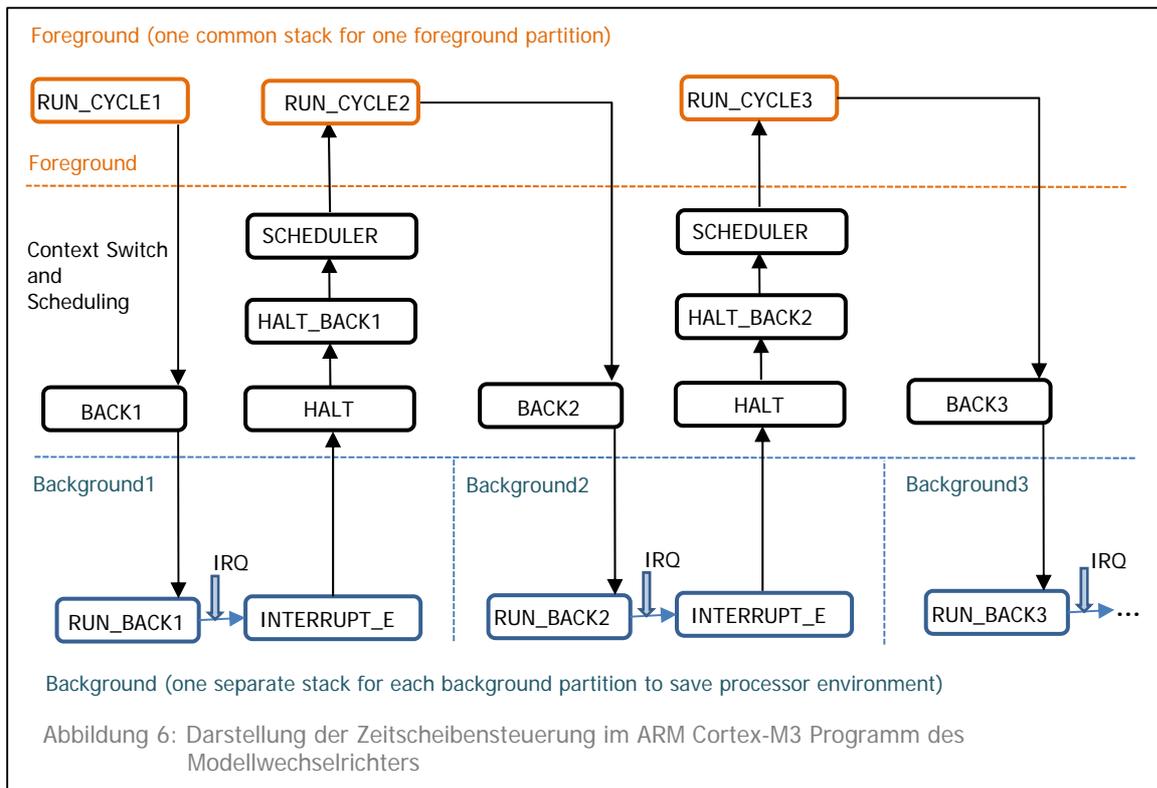


Abbildung 6: Darstellung der Zeitscheibensteuerung im ARM Cortex-M3 Programm des Modellwechselrichters

Bei der realisierten Zeitscheibensteuerung im Modellwechselrichter wurden beispielhaft eine Vordergrundpartition zur Bearbeitung der benutzerdefinierten Algorithmen und vier Hintergrundpartitionen zu Überwachungszwecken und zur Abarbeitung des Monitors umgesetzt. Der Scheduler lässt sich jedoch einfach um weitere Zeitscheiben erweitern, falls dies erforderlich sein sollte. Eine schematische Darstellung der Zeitscheibensteuerung zeigt Abbildung 6. Die Zeit vom Auftreten des IRQ bis zum Eintritt in den nachfolgenden Vordergrundprozess RUN_CYCLE_x ist in Abbildung 5 dargestellt und beträgt bei einer maximalen Kernfrequenz des Cortex-M3 (SYSCLK) von 72 MHz etwa 1.7 µs.

6 Software

Die Software des Modellwechselrichters wurde vollständig in Assembler erstellt. Aus Gründen der Codegröße und aus Validierungsgründen wurde auf die bei der Programmierung mit der Programmiersprache C häufig verwendeten Funktionen der CMSIS verzichtet. Jedoch wurden Assemblermacros und eine umfangreiche Kommentierung erstellt, um die Lesbarkeit des Codes zu erhöhen. So wurden zum Beispiel die bei vielen Assemblern für Speicherzugriffe üblichen Bitbefehle wie bset, bclr, btst und bchg eingeführt, die den zunächst schwer zu lesenden Bit Banding Mechanismus des ARM Cortex verwenden [2],[5-6].

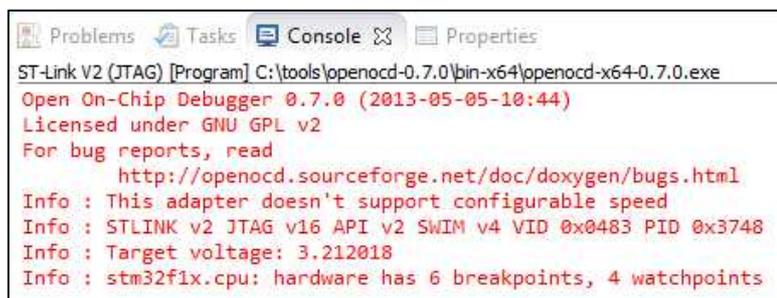
Um Studierenden eine möglichst kostenfreie Variante für weitere Untersuchungen zu geben, wurde die frei verfügbare Variante der CodeSourcery Toolchain von Mentor Graphics verwendet [14]. Diese stützt sich auf den sehr weit verbreiteten GNU ARM Assembler und hat somit keine Beschränkungen hinsichtlich Wiederverwendbarkeit oder Codegröße wie etwa die integrierte Entwicklungsumgebung MDK-ARM der Firma KEIL [15]. Letztere bietet zwar auch eine freie, aber auf 32 kByte beschränkte Version ihrer Entwicklungsumgebung samt Toolchain an. Vorteil der KEIL Entwicklungsumgebung jedoch ist, dass bereits ein Simulator integriert ist, der bei den GNU Tools fehlt.

Als IDE wurde die ebenfalls frei verfügbare und sehr weit verbreitete Eclipse-Umgebung der Eclipse Foundation eingesetzt [16]. Sie bietet den Vorteil, dass mit ihr benutzerfreundlich Übersetzungsvorgänge oder externe Programme gestartet werden können.

Gleichzeitig ist bereits eine Debugging-Umgebung integriert, die im Zusammenspiel mit dem in der CodeSourcery Toolchain enthaltenen GNU Debugger (GDB) ein komfortables on-chip Debugging ermöglicht. Hierzu muss lediglich einmalig ein on-chip Debug Server gestartet werden, der bei Auftreten von Debugging Events diese an den Debugger weiterleitet bzw. auftretende Debug Requests vom Debugger an die Hardware weiterreicht.

Für den Modellwechselrichter wurde als on-chip Debug Server das ebenfalls frei erhältliche Programm openOCD verwendet [17]. Dieses wurde von Dominic Rath im Rahmen einer Diplomarbeit an der Fachhochschule Augsburg entwickelt und kontinuierlich weiterentwickelt.

Es unterstützt zwischenzeitlich eine Vielzahl von ARM Prozessoren und Debug Adaptern. Abbildung 7 zeigt einen erfolgreichen Verbindungsaufbau mit einem ST Link/V2, der über eine JTAG Verbindung an einen STM32F103VCT6 Prozessor angekoppelt wurde.



```

Problems Tasks Console Properties
ST-Link V2 (JTAG) [Program] C:\tools\openocd-0.7.0\bin-x64\openocd-x64-0.7.0.exe
Open On-Chip Debugger 0.7.0 (2013-05-05-10:44)
Licensed under GNU GPL v2
For bug reports, read
  http://openocd.sourceforge.net/doc/doxygen/bugs.html
Info : This adapter doesn't support configurable speed
Info : STLINK v2 JTAG v16 API v2 SWIM v4 VID 0x0483 PID 0x3748
Info : Target voltage: 3.212018
Info : stm32f1x.cpu: hardware has 6 breakpoints, 4 watchpoints

```

Abbildung 7: Verbindungsaufbau mit einem ST-Link/V2 via openOCD

Die Kommunikation zwischen GDB und openOCD erfolgt dabei standardmäßig über das TCP/IP Protokoll, Port 3333 [18].

7 Algorithmen

7.1. Feldorientierte Regelung

Anhand des Modellwechselrichters sollte untersucht werden, ob die Rechenzeit im Vordergrund einer Zeitscheibe ausreichend ist, um heute gängige Verfahren zur Ansteuerung von Drehstrommotoren zu realisieren. Dazu müssen nach Wechsel in den Vordergrund im RUN_CYCLE x ($x=1\dots4$) folgende Schritte bearbeitet werden:

- Ausgabe der im letzten Zyklus berechneten PWM Kommandos
- Einlesen der Sollwerte vom CAN-Bus
- Einlesen und Aufbereiten der Messwerte
- Berechnung der Algorithmen zur feldorientierten Regelung
- Wechsel in den Hintergrund RUN_BACK x ($x=1\dots4$)

Die Algorithmen implementieren eine Umrechnung der Maschinengrößen in ein transformiertes Koordinatensystem. Das seit einigen Jahren gängige Verfahren zur Darstellung aller Feld- und Zustandsgrößen eines Motors ist die Feldorientierung, bei der alle Größen in ein meist mit dem Erregerfeld des Rotors mitrotierendes, orthogonales Koordinatensystem transformiert werden [19]. Zwar sind auch durchaus andere Ansätze möglich, jedoch hat das Verfahren der Feldorientierung den großen Vorteil, dass die bei einer Stromregelung verwendeten Sollwerte bei konstantem Drehmoment zeitlich unabhängig werden. Damit lassen sich –verglichen zum Beispiel mit dem Verfahren nach [20]– auch integrierende Anteile im Stromregler realisieren. Dadurch entfällt die Notwendigkeit von Nachlaufreglern. Nach Berechnung der Reglerausgangsgrößen müssen diese allerdings wieder mit Hilfe der inversen PARK- und CLARKE-Transformationen in das dreisträngige, statorfeste Koordinatensystem der Drehstrommaschine umgerechnet werden. Diese bilden dann die Kommandos für den im CPLD realisierten Pulsweitenmodulator.

7.2. CAN-Bus

Die Sollwerte des Modellwechselrichters werden mit Hilfe des bereits im STM32 integrierten CAN-Controllers von einem übergeordneten Kontrollrechner über den CAN-Bus vorgegeben. Um eine hohe Kompatibilität zu gewährleisten, wurde das Protokoll nach dem CANopen Standard integriert. Die Funktionalität wurde dabei auf die einer Minimum Bootup Node im Protokoll beschränkt, allerdings ohne SDO Kommunikation, siehe Abbildung 8. Nachdem die Wechselrichterparametrierung über die RS232 Schnittstelle des integrierten Monitors erfolgt, ist eine SDO Kommunikation auch nicht zwingend erforderlich, sofern PDO und NMT vollständig zur Verfügung stehen. Im Detail wurden folgende, zum Teil parametrierbare Standardwerte für die Kommunikationsparameter gewählt:

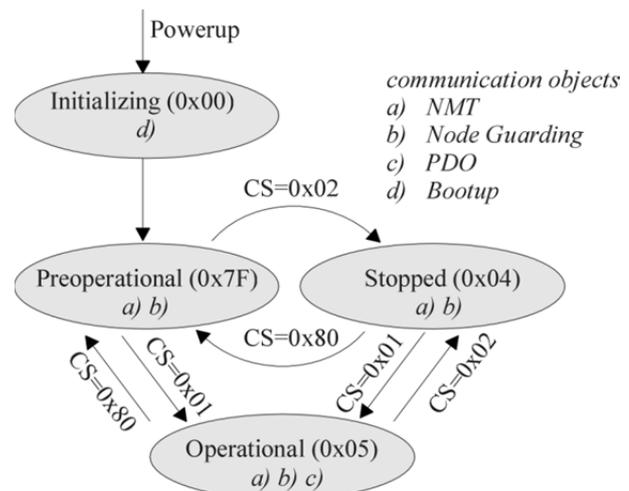


Abbildung 8: Zustandsdiagramm der CANopen-Kommunikation des Modellwechselrichters

- Baud Rate: 250 kbaud
- CAN 2.0 A identifier: 11-bit (standard identifier)
- Data Byte Format: Little Endian
- Data Format: CANopen Word Format (2 Bytes)
- Standard Node-ID: 0x03
- Sync counter: 0x01
- NMT | Sync COB-ID: 0x000 | 0x080
- Implementation Type: Slave Node with integrated NMT node guarding
- NMT States: Initializing, Preoperational, Stopped
- PDO communication: PDO1 (tx) and PDO1 (rx)
- SDO communication: missing

6.2. Hintergrund-Monitor

Der Monitor läuft in einer der vier Hintergrundpartitionen des Realtime Schedulers und wird je nach zur Verfügung stehender Rechenzeit aufgerufen. Die Ausgabe erfolgt in einem VT100 Terminal über eine serielle Schnittstelle mit den Verbindungsdaten 9600 baud, 8N1.

Abbildung 9 zeigt die Tracefunktion des Monitors, die kontinuierlich die niederwertigen 16-bit der Systemzeit (erste Spalte) als Vielfache der Dauer einer Zeitscheibe –im Modellwechselrichter 50 µs– anzeigt. Ebenso werden das System- und Benutzerstatuswort sowie 10 parametrierbare Speicherhalbworte in einer Tracetabelle angezeigt.

The screenshot shows a terminal window titled 'COM1:9600baud - Tera Term VT'. The menu bar includes 'Datei(F)', 'Editieren', 'Einstellungen', 'Steuerung', 'Fenster', and 'Hilfe'. The main area displays a trace table with the following columns: 'Time', 'SYS_Stat', 'USE_Stat', and ten columns of memory words. The table is currently in 'Trace_Hlt' mode. Below the table, four orange arrows point to specific columns with labels: 'Systemzeit' (pointing to the 'Time' column), 'Systemstatuswort' (pointing to 'SYS_Stat'), 'Benutzerstatuswort' (pointing to 'USE_Stat'), and '10xSpeicherwerte (16-bit)' (pointing to the ten memory word columns).

Time	SYS_Stat	USE_Stat	2000	2000	2000	2000	2000	2000	2000	2000	2000	2000
			B814	B816	906B	906D	9054	9056	B838	9054	B83E	901A
1B97	00000900	0000F801	1DB7	00F3	0000	0000	0305	0107	0001	0305	0047	0000
232B	00000900	0000F801	2547	00F3	0000	0000	0305	0107	0001	0305	0047	0000
2ABB	00000900	0000F801	2CDB	00F3	0000	0000	0305	0107	0001	0305	0047	0000
324F	00000900	0000F801	346F	00F3	0000	0000	0305	0107	0001	0305	0047	0000
39E3	00000900	0000F801	3BFF	00F3	0000	0000	0305	0107	0001	0305	0047	0000
4173	00000900	0000F801	4393	00F3	0000	0000	0305	0107	0001	0305	0047	0000
4907	00000900	0000F801	4B27	00F3	0000	0000	0305	0107	0001	0305	0047	0000
509B	00000900	0000F801	52B7	00F3	0000	0000	0305	0107	0001	0305	0047	0000
582B	00000900	0000F801	5A4B	00F3	0000	0000	0305	0107	0001	0305	0047	0000
5FBF	00000900	0000F801	61DB	00F3	0000	0000	0305	0107	0001	0305	0047	0000
6753	00000900	0000F801	696F	00F3	0000	0000	0305	0107	0001	0305	0047	0000
6EE3	00000900	0000F801	7103	00F3	0000	0000	0305	0107	0001	0305	0047	0000
7677	00000900	0000F801	7893	00F3	0000	0000	0305	0107	0001	0305	0047	0000
7E0B	00000900	0000F801	8027	00F3	0000	0000	0305	0107	0001	0305	0047	0000
859B	00000900	0000F801	87BB	00F3	0000	0000	0305	0107	0001	0305	0047	0000
8D2F	00000900	0000F801	8F4B	00F3	0000	0000	0305	0107	0001	0305	0047	0000

Abbildung 9: Monitor-Tracefunktion: Ausgabe von 10xSpeicherwerten (16-bit) und des Programmstatus

Mit den Tasten 4 und 5 kann auf bis zu vier weitere Tracetabellen umgeschaltet werden, so dass sich insgesamt bis zu 40 Speicherhalbworte (16-bit) darstellen lassen. Zusätzlich werden der Softwarestand, sowie die Versionsnummer der Parameterdatei zur Motorenkonfiguration dargestellt.

```

COM1:9600baud - Tera Term VT
Datei(F) Editieren Einstellungen Steuerung Fenster Hilfe
eEnd_Mem 1Set_Addr 2Addr - 3Addr + 4Read.W 5Write.W 6W_Cnt 08 7A_Inc 02

SW:*TSTVERSION* CFG:*NO CONFIG* Memory_Debug

Time  SYS_Stat  USE_Stat  2000 2000 2000 2000 2000 2000 2000 2000 2000 2000 2000
      B814 B816 907D 907F 9054 9056 B838 9066 B83E 901A

6E93  00010900  00000000  709F 03DC 0000 0000 7FFE 3FFF 0003 0300 0C00 0000
9C87  00010900  00000000  9E8F 03E1 0000 0000 7FFE 3FFF 0003 0300 0C00 0000
08CF  00010900  00000000  0ADB 03E6 0000 0000 7FFE 3FFF 0003 0300 0C00 0000

[20000000] = A22F EAFB 9D85 81C5 1554 D478 97BA 8090
[20000010] = 1BE8 608A 09B8 C79B 3DF5 4F7F 899B 079B
[20000020] = A5BA 2578 2915 7155 505E A7F8 8BE2 E538
[20000030] = █

```

Abbildung 10: Monitor-Speicherfunktion: Auslesen/Manipulieren von Speicherinhalten

Weitere Funktionalitäten sind die Modifikationsmöglichkeit von System- und Benutzerstatuswort, das Durchführen von Rücksetzfunktionen, das Hochladen von Parameterdateien im s19-Record Format mit on-chip Flashmöglichkeit, das Herunterladen von Realtime Traces von bis zu 4x16-bit Variableninhalten (mit konfigurierbarem Abtastintervall), sowie eine Auslesemöglichkeit des im Chip integrierten Speichers, siehe Abbildung 10. In flüchtigen Speicherbereichen des Prozessors können über diese Schnittstelle zusätzlich Daten manuell verändert werden.

8 Ergebnis

Im Rahmen der Untersuchungen konnte die Verwendbarkeit von ARM Cortex-M3 Kernen zum Einsatz in voll digitalisierten Wechselrichtern voll umfänglich bestätigt werden. Die erstellten Programme werden im Rahmen von Projekten an die Studierenden weitergegeben und die gewonnenen Erfahrungen in die Lehrveranstaltungen integriert werden. Eine Schnittstelle zur Lehre ist zum Beispiel das Rennwagenprojekt des Running Snail Racing Teams der Technischen Hochschule Amberg-Weiden, bei dem als Embedded Controller der Cortex-M4 Prozessor STM32F407, also ein Cortex-M3 mit zusätzlicher FPU, eingesetzt werden soll.

Es ist gelungen, eine klar strukturierte, umfassend kommentierte, übersichtliche und einfach zu validierende Software mit einer Programmcodegröße von deutlich unter 20 kByte mit dem Thumb Befehlssatz des Cortex-M3 zu realisieren. Trotz der geringen Speichergröße beinhaltet das Programm neben dem Realtime Scheduler einen vollständigen Hintergrundmonitor, eine CAN-Busanbindung nach CANopen Standard (Minimum Bootup Node ohne SDO), sowie die für Wechselrichter notwendige Parametriermöglichkeit durch Lade- und Flashfunktionen des Hintergrundmonitors.

Es konnten alle notwendigen Programmteile platzsparend aus bisher realisierten, voll digitalisierten Wechselrichtern portiert und mit Hilfe des Befehlssatzes des Cortex-M3 problemfrei integriert werden. Dabei wurden fehlende Befehle durch Macrobefehle mit Hilfe der Macrofunktionalität des GNU ARM Assemblers nachgebildet.

Zum Leistungsnachweis wurde ein vollständig in Assembler entworfener Echtzeit-Scheduler auf einem Modellwechselrichter implementiert, mit dem die Algorithmen simultan von einem Antrieb bis zu maximal vier Antrieben gerechnet werden können.

9 Abkürzungen

- AD Analog/Digital
- ARM Advanced RISC Machine
- CAN Controller Area Network
- CMSIS Cortex Microcontroller Software Interface Standard
- CTO Computing Time Overrun
- ISR Interrupt Service Routine
- JTAG Joint Test Action Group
- CPLD Complex Programmable Logic Device
- DMIPS Dhrystone Million Instructions Per Second
- FPU Floating Point Unit
- FSMC Flexible Static Memory Controller
- HDL Hardware Description Language
- IDE Integrated Development Environment
- IRQ Interrupt ReQuest
- NMT Network Management
- openOCD open On-Chip Debugger
- PDO Process Data Object
- RISC Reduced Instruction Set Computer
- SDO Service Data Object
- SNR Signal-to-Noise Ratio
- USART Universal Synchronous Asynchronous Receiver Transmitter

10 Literatur

- [1] Frenzel, B.; Kurzweil, P.; Rönnebeck, H.: Electromobility concept for racing cars based on lithium-ion batteries and supercapacitors. In: *Journal of Power Sources* 196 Elsevier (2010), S. 5364-5376
- [2] Yiu, J.: The Definitive Guide to the ARM Cortex-M3. 2nd Edition. Oxford : Elsevier, 2010.
- [3] ST MICROELECTRONICS: *STMicroelectronic STM32 32-bit ARM Cortex MCUs - STMicroelectronics*. [Online] Verfügbar unter: <http://www.st.com/web/en/catalog/mmc/FM141/SC1169> [zuletzt abgerufen am: 17.03.2014]
- [4] ST MICROELECTRONICS: *STM32F10xxx/20xxx/21xxx/L1xxxx Cortex-M3 programming manual, PM0056*. Genf, 2011 (15491). – Firmenschrift
- [5] ST MICROELECTRONICS: *STM32F101xx, STM32F102xx, STM32F103xx, STM32F105xx and STM32F107xx advanced ARM-based 32-bit MCUs, Reference Manual RM0008*. Genf, 2011 (13902). – Firmenschrift
- [6] HITEX DEVELOPMENT TOOLS: *The Insider's Guide To The STM32 ARM Based Microcontroller*. Coventry, UK, 2008. – Firmenschrift
- [7] ST MICROELECTRONICS: *STM32F103xC STM32F103xD STM32F103xE High-density performance line ARM-based 32-bit MCU with 256 to 512KB Flash, USB, CAN, 11 timers, 3 ADCs, 13 communication interfaces*. Genf, 2011 (14611). – Firmenschrift
- [8] TEKTRONIX: Tektronix Logic Analyzer, TLA6000 Series Data Sheet. Beaverton, Oregon, 2011 (25734-3). – Firmenschrift
- [9] Frenzel, B.; Kurzweil, P.; Parspour, N.: Resolver Based Position Sensing in Automotive Applications. In: *Conference Proceedings PCIM Power Conversion Intelligent Motion* (2012), S. 1639-1645
- [10] ANALOG DEVICES: Isolated Sigma-Delta Modulator, AD7401A. Norwood Massachusetts, 2011 (Rev. C). – Firmenschrift
- [11] ANALOG DEVICES: Isolated, 5 kV, DC-to-DC Converter, ADuM6000. Norwood Massachusetts, 2013 (Rev. D). – Firmenschrift
- [12] ST MICROELECTRONICS. *ST-LINK/V2 ST-LINK/V2 in-circuit debugger/programmer for STM8 and STM32*. [Online] Verfügbar unter: <http://www.st.com/web/catalog/tools/FM146/CL1984/SC724/SS1677/PF251168> [zuletzt abgerufen am 17.03.2014]
- [13] LOGILINK. *LogiLink® / PCI Express Schnittstellenkarte Parallel 1x + Seriell 2x LogiLink®*. [Online] Verfügbar unter: <http://www.logilink.eu/showproduct/PC0033.htm> [zuletzt abgerufen am 17.03.2014]
- [14] MENTOR GRAPHICS: *Sourcery CodeBench Lite Edition including ARM GCC IDE*. [Online] Verfügbar unter: <http://www.mentor.com/embedded-software/sourcery-tools/sourcery-codebench/editions/lite-edition/> [zuletzt abgerufen am 17.03.2014]
- [15] KEIL: *Keil MDK-ARM*. [Online] Verfügbar unter: <http://www.keil.com/arm/mdk.asp> [zuletzt abgerufen am 17.03.2014]
- [16] ECLIPSE FOUNDATION: *Eclipse - The Eclipse Foundation open source community website*. [Online] Verfügbar unter: <http://www.eclipse.org/> [zuletzt abgerufen am 17.03.2014]
- [17] THE OPENOCD PROJECT: *Open On-Chip Debugger*. [Online] Verfügbar unter: <http://openocd.sourceforge.net/> [zuletzt abgerufen am 04.02.2014]
- [18] THE OPENOCD PROJECT: *Open On-Chip Debugger: OpenOCD User's Guide. Release 0.8.0-dev*. [Online] Verfügbar unter: <http://openocd.sourceforge.net/doc/pdf/openocd.pdf> [zuletzt abgerufen am 04.12.2013]
- [19] Frenzel, B.; Gebhard, F.; Kurzweil, P.: Physik Formelsammlung. 3. Aufl. Wiesbaden : Springer Vieweg, 2014.
- [20] Frenzel, B.; Kilger, F.: *Method for current regulation of permanently excited synchronous motors for guided missiles having an electromechanical actuating drive for the rudder*. United States Patent and Trademark Office. April 25, 2002. US20020047680A1