Mikrocontroller ATMEL AT32UC3A USART

Bei dem **AVR32** handelt es sich um eine properitäre 32-bit-RISC-Architektur des Herstellers ATMEL. Das **EVK1100** ist ein Evaluierungs- und Entwicklungswerkzeug für den AVR32 AT32UC3A-Mikrocontroller.

IAR Systems bietet ein KickStart-Kit für Anwender des AVR32-UC-Cores von Atmel an. Enthalten sind das Entwicklungsboard EVK1100 von Atmel, eine auf 32 KByte Code limitierte Version der **IAR Embedded Workbench** für AVR32 und eine **JTAG In-Circuit Emulator mk-II** Debug-Probe von Atmel.

In der Entwicklungsumgebung IAR Embedded Workbench sind alle notwendigen Programmentwicklungs-Tools (C/C++ Compiler, Assembler, Linker, Library Builder, Editor, Projektmanager, Debugger und Simulator) integriert.

EVK1100 - Evaluation-Kit für ATMEL-AVR-UC3-Projekte

Versuchsbetreuer
Dipl.-Ing. Th. Wegner
Büro: W1314
Tel.: 498 7267

PC-Labor
Raum: W1218
Tel.: 498 7258

Aufgabenstellung
- Versuchsziel
- Grundlagen
- Studienfragen
- Aufgaben
- Literatur
- Anhang

Schnelleinstieg
- Publikationen
- Anfahrt
- Kontakt
- Laborpraktikum
- Lehrangebot
- Highlights
- Projekte

1. Versuchsziel

Es sollen die Funktion und die Assembler-Programmierung des Mikrocontrollers AT32UC3A am Beispiel der seriellen Kommunikation demonstriert werden. Zur richtigen Ansteuerung der sechs LEDs auf dem Enwicklungsboard EVK1100 ist daher die entsprechende Konfiguration der USARTs, des Interruptsystems und des GPIO-Controllers notwendig.

2. Grundlagen

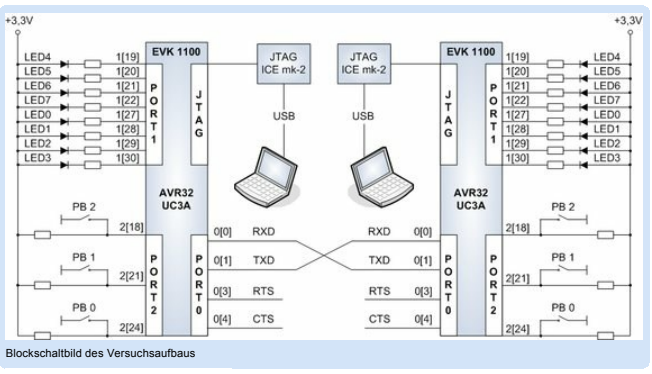
Der AVR32-RISC-Prozessors **AT32UC3A** ist ein Mitglied der AVR32-UC3-Mikrocontrollerfamilie der Firma ATMEL. Er verfügt über 512 KB internem Flash und 64 KB internem Arbeitsspeicher. Es handelt sich um einen 32 bit Prozessor mit einer Taktfrequenz von bis zu 66 MHz. Auf dem Board EVK1100 wird ein 12 MHz-Oszillator verwendent (PBA Clock=12 MHz). Aufgrund einer umfangreichen Integration von Standard-Peripherie, des flexiblen Interruptsystems, der hohen Rechengeschwindigkeit und der verschiedenen Betriebsmodi bietet sich ein breites Einsatzspektrum in eingebetteten Systemen.

Die **AVR32-CPU** besitzt einen orthogonalen Befahlssatz, daher können alle Register des allgemeinen Register-Files (Allzweckregister R0-R15) als Quelle oder Ziel bei Registeroperationen verwendet werden. Der Befehlssatz enthält kompakte (Länge 16 bit) und erweiterte (Länge 32 bit) Instruktionen. Der Stack Pointer (SP) ist in Register R13 des Register-Files gemappt. Ebenso belegen das Link Register (LR) und der Programm Counter (PC) die Register R14 und R15. Das LR-Register enthält die Rücksprungadresse eines Unterprogramms. Das PC-Register enthält die Adresse der aktuell ausgeführten Instruktion. Neben den Allzweckregistern enthält der AVR32UC noch zahlreiche Systemregister (Status- und Konfigurationsregister), die nur über spezielle Assemblerbefehle (MTSR, MFSR) gelesen oder geschrieben werden können.

Der **GPIO-Controller** (General Purpose Input Output Controller) dient dem Management der Ein/Ausgabeleitungen des Mikrocontrollers. Jeder als Eingang oder Ausgang konfigurierte E/A-Pin kann wahlweise vom PIO-Controller oder den internen Peripheriemodulen angesteuert werden.

Der integrierte Parallel-Seriell-Wandler (**USART** - Universal Synchronous/Asynchronous Receiver/Transmitter) stellt eine voll duplexfähige synchrone oder asynchrone serielle Schnittstelle mit DMA-Anbindung zur Verfügung. Duch die automatische Steuerung der Hardware Handshake Signale RTS und CTS ist eine "out-of-band" Flußsteuerung bei der seriellen Kommunikation möglich. Das USART-Modul unterstützt drei Test-Modes: Remote Loopback, Local Loopback und Automatisches Echo. Auf dem Entwicklungsboard EVK1100 sind die beiden RS232-Schnittstellen **UART_0** und **UART_1** über die genormten Steckverbinder zugänglich.

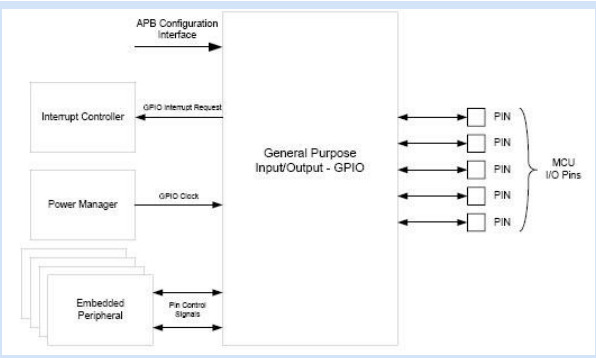
Die Programmentwicklung und Testung auf dem Mikrocontroller AT32UC3A0512 wird durch ein leistungsfähiges On-Chip Debug System (OCD) unterstützt. Dieses Debug System basiert auf dem Standard NEXUS 2.0 claas 2+ und kann über ein externes Debug Tool (JTAGICE mk-II) am JTAG-Port des **EVK1100** für den Download und Trace des Programms benutzt werden.
Die Programmentwicklung erfolgt in Assembler in der integrierten Entwicklungsumgebung **IAR Embedded Workbench** unter dem MS/Windows-Betriebssystem.



Blockschaltbild des Versuchsaufbaus

2.1 Der General Purpose Input Output Controller

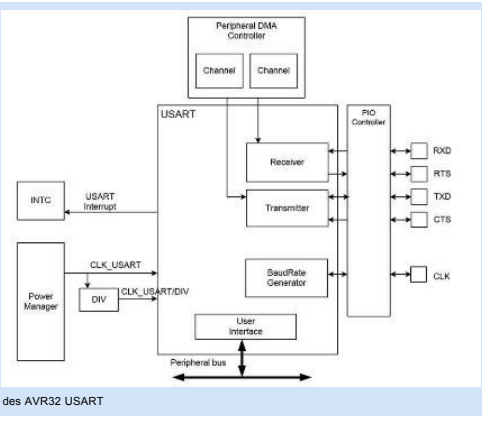
Der **GPIO-Controller** (General Purpose Input Output Controller) dient dem Management der Ein/Ausgabeleitungen des Mikrocontrollers. Jeder als Ein- oder Ausgang konfigurierte E/A-Pin kann wahlweise vom PIO-Controller oder den internen Peripheriemodulen angesteuert werden. Alle Konfigurationsregister sind innerhalb von 4 Ports organisiert. Das Register GPIO_OVR (output value register) fungiert als Output-Register. Über das Input-Register GPIO_PVR (pin value register) können digitale Eingangssignale von den Controller-Pins eingelesen werden. Bei entsprechender Konfiguration des externen Interrupt Controllers (EIC) kann der GPIO-Controller Interrupt Requests bei der CPU anmelden, die durch externe Ereignisse an den PINs (Flanken- oder Pegeländerungen) ausgelöst werden. Eine detaillierte Funktionsbeschreibung des GPIO-Controller liefert ATMEL im Kapitel 22 (GPIO-Controller) des AT32UC3A-Manuals, sowie in der Applikation Note AVR32111.



Blockschaltbild des AVR32 GPIO-Systems

2.2 Universal Synchronous/Asynchronous Receiver/Transmitter

Das **USART-Modul** stellt eine voll duplexfähige asynchrone serielle Schnittstelle zur Verfügung. Über das Baud Rate Generator Register BRGR kann in Abhängigkeit vom Peripherietakt PBA (CLK_USART) die Datenübertragungsgeschwindigkeit konfiguriert werden. Die Peripherietaktsignale sind nach einem RESET für alle vier USART-Module aktiviert, können aber über das Register PBAMASK im User-Interface des Power Manager PM bei Bedarf deaktiviert werden. Jedes UASRT-Modul besitzt ein Betriebsarten- und Steuerregister (CR und MR) zur Konfiguration des verwendeten Übertragungsprotokolls (RS232, RS485, ISO7816, IrDA, SPI). Die USART-Module sind in das Interruptsystem des AVR32 eingebunden und getrennten Prioritätsgruppen zugeordnet. Auf dem Entwicklungsboard EVK1100 sind die Signale der beiden seriellen Schnittstellen **USART0** und **USART1** über die GPIO-Funktionsgruppe A auf Port0 zugänglich.



Blockschaltbild des AVR32 USART

3. Studienfragen

- 3.1 Charakterisieren Sie kurz die seriellen Übertragungsprotokolle RS232, RS485, ISO7816, IrDA und SPI, die durch die USART-Module des AVR32 auf dem Entwicklungsboard EVK1100 konfiguriert werden können.
- 3.2 Erläutern Sie das User-Interface der USART-Module. Welche Konfigurationsmöglichkeiten bieten das Control Register CR, das Mode Register MR und das Baud Rate Generator Register BRGR für den Betrieb im asynchronen Receiver/Transmitter-Mode?
- 3.3 Wie erfolgt die Datenübertragung in dem jeweiligen seriellen Sende- und Empfangskanal?
- 3.4 Welche internen Testmöglichkeiten bieten die USART-Module des AVR32-Controllers?
- 3.5 Warum muss die Bitübertragungsgeschwindigkeit im Sender und Empfänger gleich sein?
- 3.6 Erläutern Sie die Einbindung der USART-Module in das Interruptsystem des AVR32-Controllers.
- 3.7 Erläutern Sie das Paritätsverfahren zur Fehlererkennung bei der seriellen Datenübertragung. Wie erfolgt im USART-Modul des AVR32 die Bildung und Übertragung des Parity-Bits?
- 3.8 Welche Aufgabe erfüllt das XON/XOFF-Software-Handshaking-Protokoll bei der asynchronen seriellen Datenübertragung von ASCII-Code?

4. Aufgaben

- 4.1 Erzeugen Sie in Ihrem Heimatverzeichnis einen eigenen IAR-Workspace anhand der Praktikumsvorlage.
- 4.2 Mit Hilfe der vorgegebenen Funktionsbibliothek soll ein "Hello World" auf dem 4-zeiligen LCD-Anzeigemodul des EVK1100-Boards ausgeben werden.
- 4.3 Erweitern Sie Ihr Assemblerprogramm für die Konfiguration der Peripheriemodule USART0, INTC und GPIO.
- 4.4 Schreiben Sie ein Programm zur Duplexkommunikation zwischen zwei EVK1100-Boards über die serielle Schnittstelle UART_0. Durch Betätigen der Schalter PB0-PB2 auf dem Senderboard sollen auf dem Empfängerboard die LEDs angesteuert werden und Textzeichen auf der LCD-Anzeige erscheinen.
- 4.5 Konfigurieren Sie das Interruptsystem zur Steuerung der Unterprogramme für den Sende- und Empfangsbetrieb. Welche Funktion erfüllt dabei des Interrupt Masken Register IMR des USART User Interfaces?

5. Literaturverweise

Chip AVR32

- http://de.wikipedia.org/wiki/Atmel_AVR32
- http://www.avr-asm-tutorial.net/avr_de/index.html
- ATMEL AVR32 Homepage: www.atmel.com/avr32
- AVR32 Forum Homepage: www.avrfreaks.net
- ATMEL EVK1100 Evaluation Kit
- ATMEL EVK1100 AT32UC3A0512
- ATMEL EVK1100 Schematics
- Application Note AVR32110: Using the AVR32 Timer/Counter
- Application Note AVR32101: Configuring the AVR32 Interrupt Controller
- Application Note AVR32111: Using the AVR32 PIO Controller
- ATMEL Manual AVR32 Microcontroller AT32UC3A
- ATMEL Manual AT32UC3A: 16. Interrupt Controller
- ATMEL Manual AT32UC3A: 17. External Interrupt Controller
- ATMEL Manual AT32UC3A: 22. GPIO-Controller
- ATMEL Manual AT32UC3A: 26. USART
- ATMEL Manual AT32UC3A: 31. Peripherals Timer/Counter
- ATMEL Manual AT32UC3A: 9.3 Programmiermodell
- ATMEL AVR32 Architecture Document: 2. AVR32 Programmiermodell
- ATMEL AVR32 Architecture Document: 8. AVR32 RISC Instruction Set
- ATMEL AVR32 IAR Assembler Reference Guide
- ATMEL AVR JTAG In-Circuit Emulator mk-II: Description
- ATMEL AVR JTAG In-Circuit Emulator mk-II: User Guide
- ATMEL AVR JTAG In-Circuit Emulator mk-II: User Guide - LEDs
- ATMEL AVR JTAG In-Circuit Emulator mk-II: Quick Start Guide
- ATMEL AVR32 Studio: http://www.atmel.com/dyn/products/tools_card.asp?tool_id=4116
- Application Note AVR32015: AVR32 Studio getting started
- Application Note AVR32784: AVR32 UC3 USB Device Firmware Upgrade Bootloader
- D. Tavangarian, D. Versick: Basiswissen Rechnerstrukturen & Betriebssyteme, Verlag W3I GmbH, 2008, ISBN 3937137289, 9783937137285: 5.6. Prozessorarchitekturen
- Vorlesung IEF/MD Rechnersysteme

6. Anhang

6.1 EVK1100 - Belegung der Ein- und Ausgabeports

Alle Konfigurationsregister des GPIO-Controllers sind innerhalb von 4 Ports angeordnet. (gpio_100.h)

GPIO-Port	GPIO-Adresse	GPIO-Register	Peripherie
Port0	FFFF1000	GPER	PA[0 - 30]
Port0	FFFF1004	GPERS	PA[0 - 30]
Port0	FFFF1008	GPERC	PA[0 - 30]
Port0	FFFF100C	GPERT	PA[0 - 30]
Port0	…		
Port0	FFFF1060	PVR - pin value register	PA[0 - 30]
	Port0[0]		USART0, RXD
	Port0[1]		USART0, TXD
	Port0[2]		USART0, CLK
	Port0[3]		USART0, RTS
	Port0[4]		USART0, CTS
	Port0[5]		USART1, RXD
	Port0[6]		USART1, TXD
	Port0[7]		USART1, CLK
	Port0[8]		USART1, RTS
	Port0[9]		USART1, CTS
	…		
	Port0[20]		Joy-CTR, EXTINT[8], NMI
	Port0[21]		Temperatursensor, AD[0]
	Port0[22]		Potentiometer (0-3,30 V), AD[1]
	Port0[23]		Lichtsensor, AD[2]
	Port0[25]		Joy-A, links
	Port0[26]		Joy-B, hoch
	Port0[27]		Joy-C, runter
	Port0[28]		Joy-D, rechts
Port1	FFFF1100	GPER	PB[0 - PB.31]
Port1	FFFF1104	GPERS	PB[0 - PB.31]
Port1	FFFF1108	GPERC	PB[0 - PB.31]
Port1	FFFF110C	GPERT	PB[0 - PB.31]
Port1	…		
Port1	FFFF1150	OVR - output value register	PB[0 - 31]
Port1	FFFF1154	OVRS - ovr-set	PB[0 - 31]
Port1	FFFF1158	OVRC - ovr-clear	PB[0 - 31]
Port1	FFFF115C	OVRT - ovr-toggle	PB[0 - 31]
	Port1[19]		LED4
	Port1[20]		LED5
	Port1[21]		LED6
	Port1[22]		LED7
	Port1[27]		LED0
	Port1[28]		LED1
	Port1[29]		LED2
	Port1[30]		LED3
Port2	FFFF1200	GPER	PC[0 - 5], PX[34 - 15], PX[10 - 5]
Port2	FFFF1204	GPERS	PC[0 - 5], PX[34 - 15], PX[10 - 5]
Port2	FFFF1208	GPERC	PC[0 - 5], PX[34 - 15], PX[10 - 5]
Port2	FFFF120C	GPERT	PC[0 - 5], PX[34 - 15], PX[10 - 5]
Port2	…		
Port2	FFFF1260	PVR - pin value register	
	Port2[18]		Taste PB2
	Port2[21]		Taste PB1
	Port2[24]		Taste PB0
Port3	FFFF1300	GPER	PX[4 - 0], PX[39 - 35], PX[14 - 11]

6.2 AT32UC3A - USART Register Map

Es gibt 4 USARTs im Controller auf folgenden Peripherieadressen:

- USART0: 0xFFFF1400
- USART1: 0xFFFF1800
- USART2: 0xFFFF1C00
- USART3: 0xFFFF2000

Adresse	Register USART0	Name
FFFF1400	Control Register	CR
FFFF1404	Mode Register	MR
FFFF1408	Interrupt Enable Register	IER
FFFF140C	Interrupt Disable Register	IDR
FFFF1410	Interrupt Mask Register	IMR
FFFF1414	Channel Status Register	CSR
FFFF1418	Receiver Holding Register	RHR
FFFF141C	Transmitter Holding Register	THR
FFFF1420	Baud Rate Generator Register	BRGR
FFFF1424	Receiver Time-out Register	RTOR
FFFF1428	Transmitter Timeguard Register	TTGR
FFFF1440	FI DI Ratio Register	FIDI
FFFF1444	Number of Errors Register	NER
FFFF144C	IrDA Filter Register	IFR
FFFF1450	Manchester Encoder Decoder Register	MAN
FFFF14FC	Version Register	VERSION

6.3 AT32UC3A - Interrupt Request Signal Map

Innerhalb des Interrupt-Controllers (INTC) gibt es Interrupt-Prioritäts-Register (IPR), Interrupt-Request-Register (IRR) und Interrrupt-Cause-Register (ICR). Das Register EVBA (Exception Vector Base Address) innerhalb der AVR32-Systemregister enthält die Anfangsadresse der Interrupt-Vektor-Tabelle. Die Interrupt-Bits GM (Global Interrupt Mask) und EM (Exception Mask) befinden sich im Status Register der AVR32-CPU. Das Register ECR (Exception Cause Register) innerhalb der AVR32-Systemregister enthält den Adressen-Offset des letzten aufgetretenen Events. (intc.h)

Adresse	Register	Gruppe	Line	Module	Signal
FFFF0800	IPR0	0	0	AVR32 UC	
FFFF0804	IPR1	1	0	External Interrupt Controller	EIM0
FFFF0808	IPR2	2	0	GPIO-Controller	GPIO 0
FFFF0808	IPR2	2	1	GPIO-Controller	GPIO 1
FFFF0808	IPR2	2	2	GPIO-Controller	GPIO 2
FFFF0808	IPR2	2	…	GPIO-Controller	…
FFFF0808	IPR2	2	13	GPIO-Controller	GPIO 13
FFFF0810	IPR5	5	0	USART 0	USART0
FFFF0814	IPR6	6	0	USART 1	USART1
FFFF0818	IPR7	7	0	USART 2	USART2
FFFF081C	IPR8	8	0	USART 3	USART3
		…			
FFFF0838	IPR14	14	0	Timer/Counter	TC 0
FFFF0838	IPR14	14	1	Timer/Counter	TC 1
FFFF0838	IPR14	14	2	Timer/Counter	TC 2
FFFF083C	IPR15	15	0	Analog/Digital-Wandler	ADC
	…				
	IPR63				