Mikrocontroller ATMEL AT32UC3A Timer Counter Interrupt

Mit dem AVR32 Mikrocontroller ist eine programmierbare 32 Bit RISC-Architektur des Herstellers ATMEL entstanden. Die Firmware wird mit der Entwicklungsumgebung für den AVR32 AT32UC3A Mikrocontroller

EVK1100 - Evaluation Kit for ATMEL AVR UC3 Projects

IAR Systems bietet ein Kickstart-Kit für Anwender des AVR32UC3 Cores von Atmel an. Dieses Kickstart-Kit und die Entwicklungsumgebung IAR Embedded Workbench ermöglichen die Entwicklung und Fehlersuche mit IAR C/C++ Compiler, Assembler, Linker, Library Builder, Editor, Projektmanager, Debugger und Simulator) integriert.

1. Versuchziel

Es sollen die Funktion und die Assembler Programmierung des Mikrocontrollers AT32UC3A am Beispiel eines im Selbstschlaf gesteuerten Binärzählers demonstriert werden. Zur richtigen Programmierung des AVR32 UC3A Mikrocontrollers ist das Verstehen der Interrupt Verarbeitung und der entsprechende Konfiguration der Timer/counter, des Interrupt controllers und des GPIO-Controllers notwendig.

2. Grundlagen

Der AVR32-RISC-Prozessor **AT32UC3A** ist ein Mitglied der AVR32 UC3 Mikrocontrollerfamilie der Firma ATMEL. Es verfügt über 512 KB interen Flash und 64 KB internen Speicher. Es stehen 66 programmierbare I/O Pins zur Verfügung. Der Mikrocontroller **AT32UC3A** hat dem Board EVK1100 und ein 12 MHz Oszillator verwendet. PWA Clock=12 MHz, Ja, Adjust und unterspezifiziert Integration von Standard-Peripherie, das flexiblen Interrupt-System, der hohen Rechengeschwindigkeit und der verschiedenen Betriebsmodi bietet sich ein breites Einsatzspektrum für eingebettete Anwendungen.

Das Mikrocontrollersystem **ATMEL AT32UC3A** bietet einen umfangreichen Eventhandling-Mechanismus zur ereignisgesteuerten, hardwaremäßigen Programmverarbeitung. Mögliche Events sind Interrupts, Peripheral Events und Peripheral DMA Requests. Die Verarbeitung der Ereignisse erfolgt ohne Prozessorkenntnisse. Die Unterbrechungsverarbeitungen können je nach Event-Typ priorisiert werden, um ein deterministisches Verhalten im Fall parallel auftretender Ereignisse sicherzustellen. Wenn eine Unterbrechungsverarbeitung auftritt, wird die Abarbeitung des aktuellen Programmes unterbrochen und an der Stabadresse einer zugehörigen Serviceroutine fortgesetzt.

Die **AVR32-CPU** besitzt einen orthogonalen Befehlssatz, dahe ermöglichen alle Register des allgemeinen Register-Files (Allzweckregister R0-R15) als Quelle oder Ziel bei Registeroperationen verwendet werden. Die Speicherzugriffe können mit Byte, Halbwort und Wort erfolgen. Die Stack Pointer (SP) ist in Register R13 des Register-Files gemapt. Ebenso wird das Link Register (LR) und der Programm Counter (PC) die Register R14 und R15. Beta LR-Register enthält die Rücksprungadresse bei Subroutinen-Aufrufen und bei Interrupts. Die Rücksprungadresse wird ausgeführten Instruktion. Neben den Allzweckregistern enthält die 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 Ein- oder Ausgang konfigurierte Pin kann wahlweise von PIO-Controller oder den internen Peripheriemodulen angesteuert werden. Alle Ein/Ausgabeleitungen sind internalt über 4 Ports organisiert. Das Register PVR (PVR output value register) kontrolliert als Output-Register. Über das Input Register GPIO_PVR (pin value register) können die Pin-Zustände gelesen werden. Die Konfigurationsregister (Pin Configuration Register) dient zur 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 können. Die Konfigurationsregister (Pin Configuration Register) PER, OER, PER, PER, P