

Debugging

Den Flash-Speicher überlistet

27. Oktober 2011, 12:01 Uhr | Von Dirk Akemann

Hardware-Breakpoints sind auch bei modernen Mikrocontrollern Mangelware, sind aber für ein performantes Debugging unverzichtbar. Denn der Flash-Speicher, aus dem heraus die Programme ausgeführt werden, lässt sich für Software-Breakpoints zu langsam programmieren. Mit einem Trick klappts aber trotzdem.



Anzeige

Powermanager LTC2971

Verwaltung von Fehlern in Echtzeit,
präzise Einhaltung der 0,25 %
Lastpunktgenauigkeit



Mikrocontroller haben nur kleine RAM-Speicher und führen die Applikation in der Regel direkt aus dem Flash-Speicher heraus aus. Daher müssen Debugging-Werkzeuge auch im Flash funktionieren.

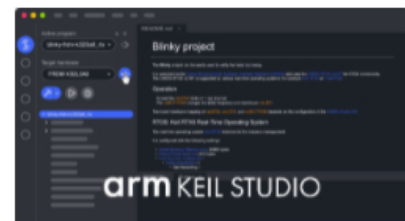
Ist der Entwickler gewohnt, beliebig viele Breakpoints in seiner Applikation zu setzen, wie es im RAM-Speicher ohne Einschränkungen möglich ist, so muss er beim Debuggen im Flash-Speicher einige Beschränkungen beachten.

Eine Lösung, diese Einschränkungen zu überwinden, sind Hardware-Breakpoints. Diese werden nicht durch Ersetzen einer Instruktion im Speicher ausgelöst, sondern durch ein Vergleichsregister, das den Wert des Program Counters vergleicht, um dann entsprechend den Prozessor zu stoppen und die Steuerung an den Debugger zu übergeben. Selbst moderne Prozessoren wie der Cortex-M3/M4 stellen nur sechs dieser Hardware-Breakpoints zur Verfügung. Beim Cortex-M1 sind es vier und bei den immer noch häufig eingesetzten ARM7- und ARM9-Prozessoren sind es sogar nur zwei.

Diese sind schnell aufgebraucht. Der Debugger selbst benötigt schon einen Breakpoint, um die Terminalausgaben zu steuern. Hinzu kommen beim Single Stepping weitere Breakpoints abhängig von der Zahl der Verzweigungen, über die gesteppt wird.

Sind die verfügbaren Breakpoints aufgebraucht, wird das Debuggen erheblich eingeschränkt. Einer der Faktoren ist dabei die Ausführungszeit, die Single Steps im Source-Code der Hochsprache benötigen. Sind keine Breakpoints mehr vorhanden, wird auf Maschinensprache-Level gesteppt, so dass bei komplexeren Operationen Single Stepping auf Source-Ebene unmöglich wird.

Themenwelt



Cloud-Tools für Arm Mikrocontroller

Keil Studio Cloud für Embedded-Projekte mit Zero-Installation, Git-Integration und Web-Debugging.

Advertorial



TQ-Group

Embedded-Module für HMI-Entwicklungen

Produkte auf HMI-Basis benötigen Embedded-Module, die den Prozessor optimal einsetzbar machen

Grenzen überwinden

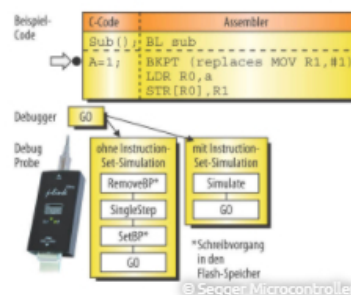
Um diese Grenzen zu überwinden, gibt es Flash-Breakpoints. Das sind Software-Breakpoints, die in den Flash-Speicher programmiert werden, um die Limitierungen der Hardware-Breakpoints aufzuheben. Doch Flash-Breakpoints sind nicht frei von Verzögerungen, da Flash abhängig von der Blockgröße Zeit benötigt, um programmiert zu werden. Um die Anzahl der Zugriffe auf den Flash-Speicher zu verringern, werden die Befehle gesammelt und dann gemeinsam geschrieben, so dass wirklich nur dann, wenn es absolut nötig ist, auf den Flash-Speicher zugegriffen wird.

Mit der „Instruction Set Simulation“ (**Bild**) geht das Unternehmen Segger noch einen Schritt weiter und verbessert die Ausführungsgeschwindigkeit beim Debuggen im Flash-Speicher. Um zu vermeiden, dass etwa beim Single Stepping ein bereits bestehender Breakpoint gelöscht und erneut gesetzt werden muss, wird die durch den Breakpoint ersetzte Maschineninstruktion simuliert.

Hierzu werden je nach Instruktion Speicherbereiche oder Register sowie der Program Counter direkt von der Debug Probe verändert, so dass das Ergebnis identisch mit der tatsächlichen Ausführung der Instruktion ist.

Viele Instruktionen, die sich nicht simulieren lassen, können im RAM-Speicher des Mikrocontrollers emuliert werden. Hierzu wird die Instruktion ins RAM kopiert. Die nächste Instruktion im RAM ist die Breakpoint-Instruktion. Anschließend wird der Program Counter auf das RAM umgestellt und die Instruktion dort ausgeführt.

Sobald der Breakpoint im RAM auslöst, wird der Program Counter an die entsprechende Stelle im Flash-Speicher zurückgesetzt. Mit diesem Verfahren lassen sich viele unnötige Flash-Zugriffe vermeiden und damit die Arbeitsgeschwindigkeit beim Debuggen spürbar erhöhen.

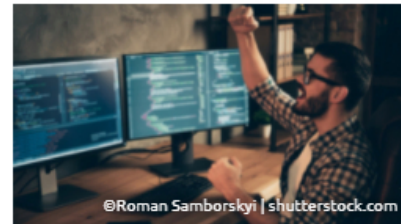


Instruction-Set-Simulation. Single-Step im Quellcode über einen vorher gesetzten Breakpoint. Die Debug Probe kann den Mikrocontroller auf verschiedene Arten manipulieren. So lassen sich z.B. Register verändern, ohne dass die CPU aktiviert werden muss.

Entwicklungen

Produkte auf HMI-Basis benötigen Embedded-Module, die den Prozessor optimal einsetzbar machen

Online-Kompodium



Embedded-Linux-Systeme tracen

Wie man den Linux-Support von Tracealyzer v4.4 von Perceptio am besten gestaltet.

Meistgelesen



Zuverlässigkeit von SSDs

Royal Flash

Konferenz »Internet of Things«

Programm ist online

Cybersecurity

Schutz vor Hackerattacken überfordert viele Firmen noch