

Optimierung des Energieverbrauchs eingebetteter Software

Welchen Einfluss hat eine Programmänderung auf den Energiebedarf einer Applikation? Welcher Programmteil verursacht den größten Energieverbrauch? Bei der Entwicklung von batteriebetriebenen Geräten wie Mobiltelefonen kommt diesen Fragen elementare Bedeutung zu. War es bisher sehr aufwendig, wenn nicht unmöglich, Strommessungen bestimmten Programmstellen zuzuordnen, so wird es mit der hier vorgestellten Messanordnung zum Kinderspiel. Der Entwicklung energieoptimierter Software steht damit nichts mehr im Wege.

Kenndaten wie Standby-Zeit oder Betriebszeit können entscheidend für den Erfolg eines Produktes sein. Aus diesem Grund werden heute beim Design batteriebetriebener Geräte umfassende Maßnahmen zur Reduktion des Energieverbrauchs ergriffen. Typische Maßnahmen sind:

- Einsatz stromsparender Bauteile mit Power Saving Modes
- Hoher Integrationsgrad
- Variation der CPU-Frequenz
- Variation der Versorgungsspannung
- Einsatz von Microcontrollern mit Power Saving Features, Cache und On-Chip Speicher

Eine optimale Reduktion des Energieverbrauchs kann allerdings nur dann gelingen, wenn die Software, die das Gerät steuert, alle Sparmöglichkeiten der Hardware konsequent ausnutzt. Die Energie ist das Produkt aus Strom, Spannung und Zeit. Jeder dieser Parameter kann durch die Steuer-Software beeinflusst werden.

Während des Entwicklungsprozesses muss deshalb immer wieder untersucht werden, ob die drei Parameter abhängig vom Betriebsmodus der Anwendung optimal aufeinander abgestimmt sind.

Es gilt beispielsweise zu überprüfen:

- ob der Microcontroller im richtigen Power Saving Mode ist
- wie sich Programmänderungen auf den Stromverbrauch auswirken
- ob unerwartete Stromspitzen auftreten

Eine solche Überprüfung erfordert eine Messanordnung, die neben dem Programm- und Datenfluss der Steuersoftware zeitgleich den Strom- und Spannungsverlauf misst, aufzeichnet und anschließend auf einfache Weise in Bezug bringt.

Anforderungen

Um den Energieverbrauch für jeden Punkt des Programms angeben zu können, müssen folgende Messdaten erfasst werden:

- Der zeitliche Programmablauf der Steuersoftware; dazu wird heute üblicherweise ein Debugger mit einem dazugehörigen Real-Time Trace verwendet.
- Der Strom- und Spannungsverlauf während der Programmlaufzeit; als Messgerät bietet sich hier ein LogicAnalyzer mit einer Analog/DigitalProbe an.

Die eigentliche Herausforderung besteht darin, die Messung des Programmflusses mit der Messung von Strom und Spannung zu korrelieren. Beim Einsatz nichtintegrierter Mess-Hardware, z.B. Einzelgeräte unterschiedlicher Hersteller, ist dies grundsätzlich mit erheblichem Aufwand verbunden, oftmals sogar unmöglich. Ebenso ist die Analyse und statistische Auswertung schwierig und wenig komfortabel.

Lösung

Bei der integrierten Messanordnung von LAUTERBACH aus Debugger, Real-Time Trace und Logic Analyzer ist jedes dieser Geräte mit einem Zeitzähler ausgestattet. Zu Beginn der Messung werden diese Zeitzähler synchronisiert. Jede einzelne Aufzeichnung wird mit einem Zeitstempel versehen und kann deshalb mit den Aufzeichnungen der jeweils anderen Messgeräte korreliert werden. Damit lässt sich exakt darstellen, wie viel Leistung zu einem bestimmten Zeitpunkt der Programmausführung von der Applikation verbraucht wurde.

Zur Ermittlung des Programmablaufes kommt ein Realtime-Trace zum Einsatz der zum Beispiel das ETM- oder NEXUS-Traceprotokoll aufzeichnet.

Zur Ermittlung des Strom- und Spannungsverlaufes kommt ein Logikanalysator mit AnalogProbe zum Einsatz. Diese ermöglicht die Echtzeitmessung von bis zu 4 Spannungskanälen und 3 Stromkanälen. Des Weiteren ist die Echtzeittriggerung auf Spannungs-, Strom- und Leistungswerte unterstützt.

Für die Strommessung muss in der Versorgungsleitung der Applikation ein Shuntwiderstand vorgesehen werden. Die über diesen Widerstand abfallende Spannung wird von der AnalogProbe gemessen. Dabei ist die gemessene Spannung abhängig vom Widerstandswert und proportional zum Strom. Diese Messmethode ist sehr verbreitet und wird von vielen Evaluierungsschaltungen unterstützt.

Strom- und Spannungskanäle und die Werte der Shuntwiderstände werden in einem Eingabefenster aktiviert (s. Abb. 1).

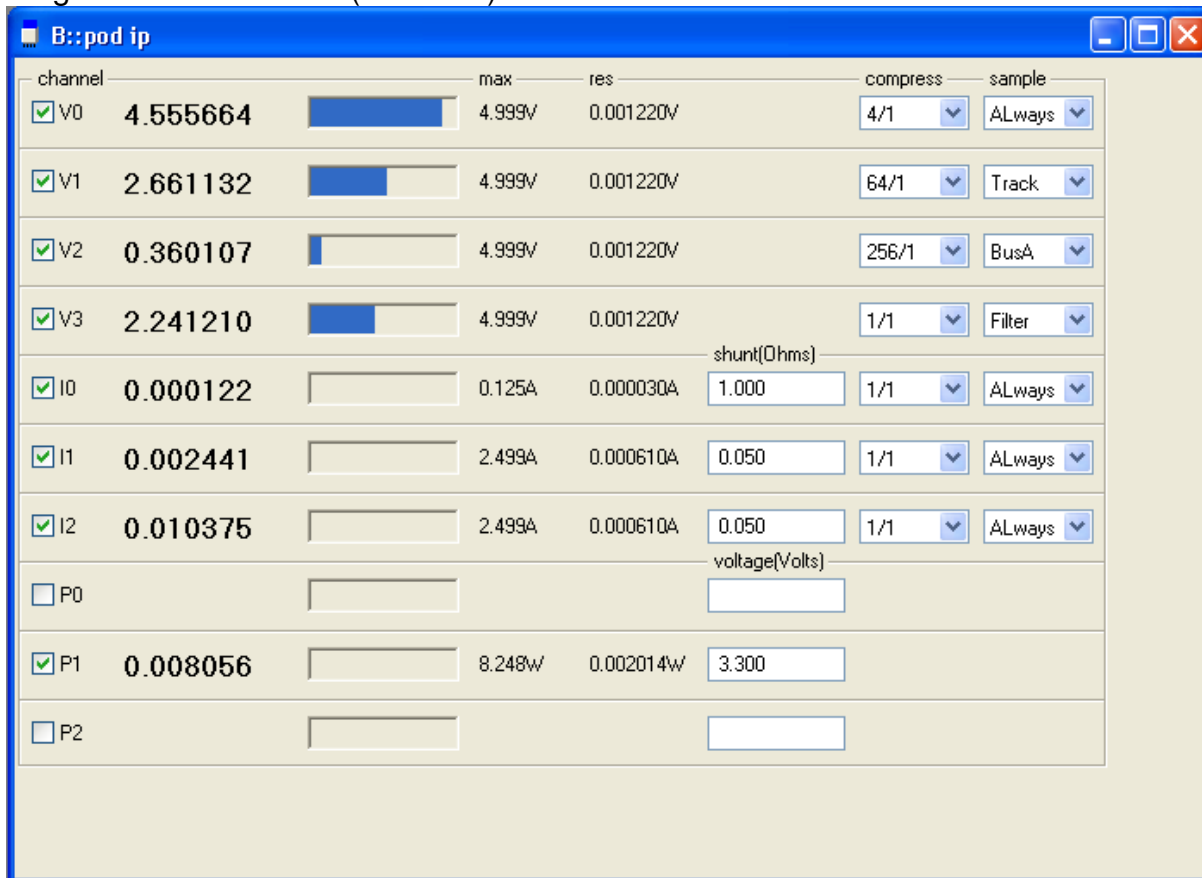


Abb.: 1

Gleiches gilt für die Leistungskanäle, deren Berechnung aus dem gemessenen Strom und der gemessenen Spannung erfolgt. Alternativ kann auch eine feste Spannung für die Berechnung angegeben werden.

Strom und Spannung lassen sich ebenso wie die daraus berechnete Leistung als zeitabhängige Kurven oder Tabellen darstellen. Alle zeitabhängigen Darstellungen (Programmfluss, Chart, Strom, Spannung, Leistung) werden durch die Track-Funktion synchronisiert. Ein 'Maus-Klick' in eines der Fenster und alle anderen Darstellungen werden automatisch nachgeführt. Durch diese enge Kopplung lassen sich die Ursachen einer hohen Leistungsaufnahme schnell der entsprechenden Programmstelle zuordnen.

In dem unten gezeigten Beispiel (s. Abb. 2) werden nacheinander 8 Leuchtdioden eingeschaltet und anschließend wird eine Sinusspannung erzeugt. In der Strom-Zeit-Kurve (1) ist der treppenförmig ansteigende Stromverbrauch dargestellt. Die Sinusspannung ist in der Spannungs-Zeit-Kurve (2) zu erkennen. In der Strom- (1), Spannungs- (2) und Chartdarstellung (3) ist die Funktion 'LED3_on' durch die senkrechte, schwarze Track-Linie markiert.

Im 'Trace.List'-Fenster (4) ist die entsprechende Programmzeile durch einen blauen Balken gekennzeichnet.

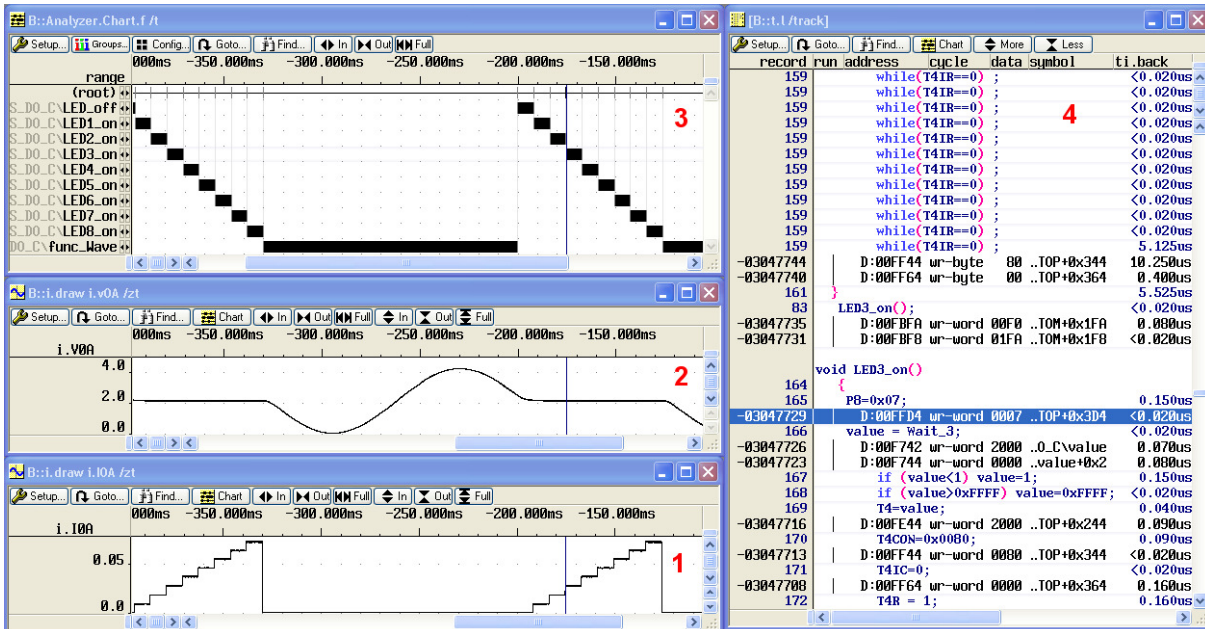


Abb.: 2

Statistische Auswertungen werden nach jedem Programmstopp automatisch ausgeführt. Sie liefern Informationen über Minimal-, Maximal- und Mittelwert des Energieverbrauches der ausgeführten Funktionen (s. Abb. 3). Ebenso wird für jede Funktion der absolute und prozentuale Anteil am Gesamtenergieverbrauch ermittelt. Energiehungrige Programmteile können damit sehr leicht erkannt werden. In ihnen steckt das größte Potential zur Energieeinsparung.

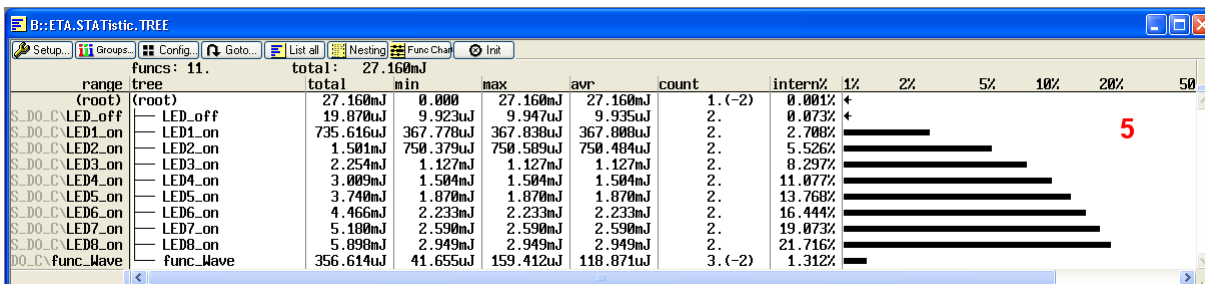


Abb.: 3

Die statistische Auswertung zeigt den Energieverbrauch der einzelnen Funktionen. Im Balkendiagramm (5) ist zu erkennen, dass die Funktion 'LED8_on' (alle LEDs an) den höchsten Energieverbrauch hat.

Als weitere Funktion steht eine Triggereinheit zur Verfügung. Durch die Angabe von Grenzwerten oder Bereichen für Strom, Spannung und Leistung werden Triggerereignisse definiert. Deren Auftreten wird von der Triggereinheit dazu benutzt, die Aufzeichnung gezielt ein- und auszuschalten (selektiver Trace). Alternativ kann auch ein Triggersignal erzeugt werden, das die Programmausführung stoppt.

Zusammenfassung

Die integrierte Messanordnung für die Energieanalyse eröffnet dem Entwickler eine einfache und schnelle Möglichkeit, Zusammenhänge zwischen einzelnen Programmpunkten und der zugehörigen Strom- bzw. Leistungsaufnahme zu erkennen. Durch die einfacher Handhabung der Messanordnung und Integration in eine bekannte Debug Bedienoberfläche ist Energieanalyse ab sofort nicht mehr nur ein Thema für Spezialisten. Jedem Entwickler ist es damit möglich effektiv den Energiebedarf einer Applikation zu optimieren.

Elmar Stahleder, Juni 2007