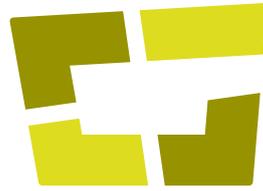


WIR ENTWICKELN PRÄZISION



GLIWA
embedded systems



TIMING 1. KLASSE

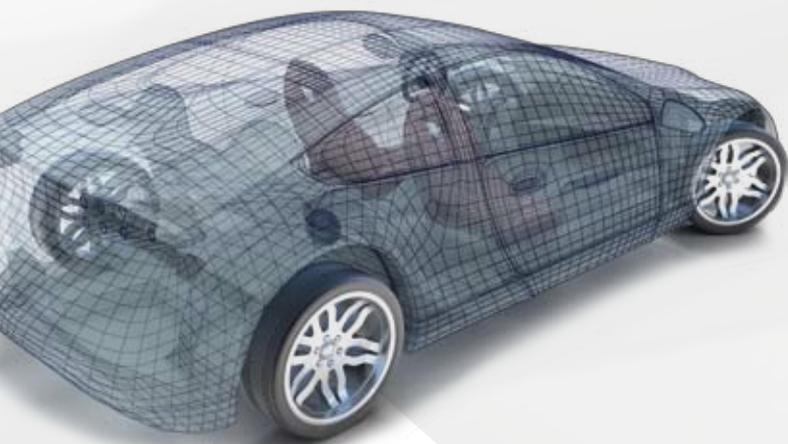
TIMINGSUITE FÜR ECHTZEITSYSTEME



TIMING 1. KLASSE

T1 - DIE UMFASSENDE LÖSUNG ZUR LAUFZEIT-MESSUNG

T1 ist die umfassende Timingsuite zur Analyse des Laufzeitverhaltens von Echtzeitsystemen. Die durchdachte Darstellung macht komplexe Abläufe in der Software leicht verständlich. Das Debugging wird optimal unterstützt. Die Ermittlung der verschiedensten Timingparameter erlaubt eine detaillierte Analyse. Bei Bedarf kann die Auswertung durch T1 Komponenten auf dem Zielsystem vorgenommen werden. So löst T1 neben reinen Messtechnikaufgaben auch die Laufzeitanalyse. Die Einhaltung der Timing-Constraints werden damit sicher nachgewiesen und bei Verletzung klar identifizierbar.



ANALYSE & VERSTÄNDNIS DES LAUFZEITSYSTEMS

T1 erlaubt Systemeinsichten. Abläufe werden sichtbar und damit verständlich. Laufzeitprobleme werden rechtzeitig erkannt und durch deren Visualisierung lösbar.

T1 IST MODULAR

T1 besteht aus unterschiedlichen Komponenten und gibt dem Anwender verschiedene Analysewerkzeuge an die Hand. Bei steigender Komplexität der Systeme und der zu bewältigenden Analyseaufgaben wächst T1 mit. So werden wichtige Meilensteine in der Entwicklungsphase abgesichert. Eine aufgrund von Laufzeitproblemen drohende Verschiebung des „SOP“ wird durch T1 verhindert.

T1 WÄCHST MIT

Bis zur Fertigstellung eines Seriensteuergeräts entstehen viele SW-Stände. Nachträgliche Erweiterungen oder Änderungen stehen an der Tagesordnung. T1 begleitet die Systemanalyse von Anfang an und zeigt frühzeitig Engpässe auf. Es entsteht ein Optimierungsprozess, der zum verlässlichen Begleiter durch alle Projektphasen wird.

VOM PILOT ZUM SERIENSTEUERGERÄT

T1 ist da, wo es benötigt wird! Ob mit dem Modultest beginnend und dann in der automatisierten Testumgebung (Bsp. HIL) oder am Ende im fertigen Seriensteuergerät: T1 erzeugt exakte Laufzeitmessungen in jeder Projektphase - sogar im fertig verbauten Steuergerät. Dabei können Messdaten zum Beispiel über CAN mit niedriger Anforderung an die Übertragungsbandbreite transferiert werden.

OPTIMIERUNG

Ein häufiger Fall: das Design ist funktional perfekt. Der Prototyp hingegen zeigt trotz enormer CPU-Reserven Timing-Verletzungen! T1 zeigt sofort, wo angesetzt werden muss. Bei der Optimierung eines hoch ausgelasteten Systems hilft T1, Optimierungsansätze zu finden und Maßnahmen zu bewerten.

DEBUGGING

Die Erfahrung hat es bewiesen: bei der Untersuchung von Timingproblemen wird in der Regel an der falschen Stelle, nämlich beim funktionalen Verhalten angesetzt. Ursache und Auswirkung eines Timingfehlers liegen jedoch oft weit auseinander. Ein „Durchhängeln“ per Debugger kann ohne weiteres Wochen veranschlagen. T1 trägt hier entscheidend dazu bei, die Fehlersuche zum Teil drastisch zu verkürzen.

DOKUMENTATION

Standards wie die ISO 26262 oder die DO 178-B fordern eine klare und durchgängige Dokumentation. T1 erstellt auf Wunsch ganz automatisch Reports des Timingverhaltens. Einmal konfiguriert, erhält der Benutzer auf Knopfdruck die benötigten Dokumente zur weiteren Verarbeitung.



T1 SICHERT «SOP» FÜR SERIENSTEUERGERÄTE

T1 begleitet alle Phasen der Laufzeitverifikation. Wirklich bekannt wurde T1 durch den Einsatz bei Seriensteuergeräten, wenn bei der Integration verschiedene Softwareanteile zusammen kommen und das System auf dem Prüfstand steht.

T1 INTEGRATION IN DAS KUNDENPROJEKT

Die GLIWA GmbH überlässt nichts dem Zufall und integriert T1 direkt beim Kunden:



- **T1-Adaption:** T1 wird an die Gegebenheiten wie Prozessor, RTOS, Compiler und Schnittstellen adaptiert. Viele Varianten stehen bereits „out-of-the box“ zur Verfügung (siehe Datenblatt auf der Rückseite).
- **T1-Integration:** T1 wird direkt in das Kundenprojekt eingepasst. Mittels Engineering-Services des GLIWA Expertenteams werden diese Anpassungen durchgeführt. Das bedeutet für den Kunden garantiertes Plug & Play.
- **T1-Nutzung:** Von jetzt an wird mit T1 gemessen.

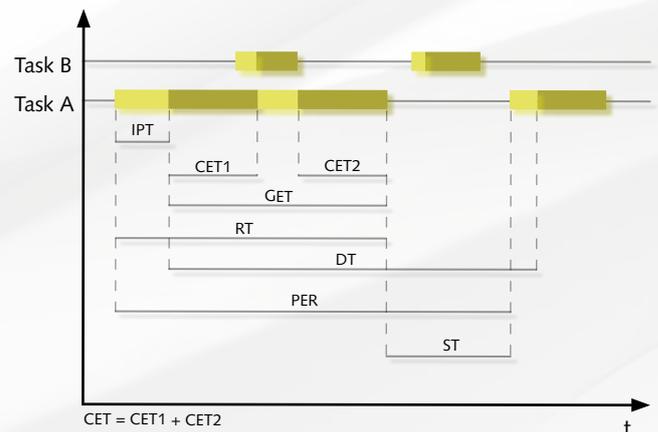
T1 EINSATZ IM MULTI-PROZESSORSYSTEM

T1 wird bereits heute in Systemen mit mehreren Kernen eingesetzt. Dies gilt für Systeme mit mehreren CPUs oder auch MultiCore Architekturen. T1 wird kontinuierlich um neue Funktionen für MultiCore Systeme erweitert.

KENN- UND MESSGRÖSSEN VON T1

Diese Tabelle gibt einen Überblick über die gängigen Zeitwerte im Bereich der Laufzeitanalyse:

| ABK. | BEDEUTUNG (ENGLISCH) | BEDEUTUNG (DEUTSCH) |
|------|----------------------|---------------------|
| IPT | initial pending time | Initialwartezeit |
| CET | core execution time | Nettolaufzeit |
| GET | gross execution time | Bruttolaufzeit |
| RT | response time | Antwortzeit |
| DT | delta time | Deltzeit |
| PER | period | Periode |
| ST | slack time | Restzeit |
| CPU | CPU load | CPU Auslastung |
| JIT | jitter | Jitter |





GRUNDFUNKTIONEN

| | |
|----------------------|--|
| Projekt-Explorer | 2 Views, System- und Applikationsview, inkl. Code-Viewer |
| Projekt-Import | Source, Object Code wie Bsp. ELF-Daten inkl. Debug-Daten |
| HTML-Reports | Reports aller Timinginfos mit Bild bei wählbarem Font / Farbe |
| CSV-Reports | formatiert, Ws, XXms, YYus, ZZns, Zeiten: ns, µs, ms, s |
| Report-Konfiguration | Frei konfigurierbar inkl. Zusammenfassung mit Grafik |
| Report-Sicherung | Ja, Projekte & Konfiguration in T1-Projektdateien |
| Symbol-Resolver | Ermittelt aus Adressen, Symbole und Debug-Informationen |
| Symbol-Search | Freie Suche nach Symbolen |
| ELF Info-Dialog | Dialog der Daten aus der ELF-Datei |
| Connection Logger | Protokolliert alle T1-CAN Botschaften (RX/TX) |
| Send Traces | Paketierte alle Daten in ZIP und generiert eine Email mit Inhalten |

MIT T1 IMMER UP TO DATE

T1 entwickelt sich ständig weiter. Das Datenblatt ist daher nur ein Überblick. T1 kann generell an jedes Projekt mit beliebiger Umgebung (CPU, RTOS, Toolumgebung, HW-Schnittstellen etc.) angepasst werden.



HOCH FLEXIBLE ON-TARGET LAUFZEITMESSUNG

T1.flex ist das Methodenwerkzeug unter anderem für vollkommen unerwartete Ereignisse, bei denen zur Laufzeit Messpunkte gesetzt werden können. Damit lassen sich ohne vorherige Instrumentierung am Code Messungen durchführen. Das bedeutet, dass die SW für den Test weder neu kompiliert noch geflashed werden muss! T1.flex nutzt spezielle Mechanismen der Controller und deren HW-Architektur. Beim Auftreten von Laufzeitproblemen können so ad hoc Messergebnisse generiert werden. Dieses Verfahren macht T1 zu einem mächtigen und hoch flexiblen Analysesystem, das besonders in kritischen Projektphasen seine Stärken unter Beweis stellt.



DAS SW-OSZILLOSKOP

T1.scope ist die Basiskomponente. Ähnlich einem Oszilloskop erfolgt die zeitliche Darstellung der Tasks grafisch auf der Zeitachse. Das Auftreten von Interrupts oder jegliche Art von Events wird ebenso visualisiert. Der Fokus der Messung kann frei konfiguriert werden.

Die Überwachung des Zugriffs bestimmter Daten ist ebenfalls möglich. Mit wenigen Mausklicks lassen sich gewünschte Bereiche vermessen. Die Datenaufbereitung geschieht im Hintergrund und bündelt sie zu relevanten Ergebnissen. T1.scope bietet viele Zusatzfunktionen. Messergebnisse lassen sich beispielsweise u. a. als html-Dateien exportieren und Projektpartner per Email zur Beurteilung (grafisch wie quantitativ) übermitteln.

FUNKTIONEN

| | |
|-------------------------------|---|
| Generell | Synchronisierte Views mit auto-update über verschiedene Fenster |
| Trace View (Grundfunktionen) | Grafische Darstellung von Laufzeitevents mit Scheduling, Stopwatches und Datenflüssen, User-Events, Constraints und Highlighting bei Constraint Violations, Overhead-Messung, Min./Max. Werte, CPU-Last Diagramm, Kommentare/Bookmarks, CPU-Last für Tasks und Gesamtsystem inkl. BSF, weitere Kommentare/Bookmarks sowie User-Time-Grids |
| Trace View (Zusatzfunktionen) | Zoomfunktion, markieren, skalieren, selektieren, Look & Feel, verschiedene Baumdarstellungen usw. |
| Speichern & Drucken | Binärtraces, T1.flex, Texttraces alle auch mit Zusatzinfo |
| Messwerte | CET, GET, IPT, RT, DT, PER, ST, PRE, JIT, CPU-Load, Occurance |
| Charts | Pie-Chart, Histogramme, Bars, statistische Darstellung inkl. extrapolierter Werte |
| Korrekturen | Korrekturen und Kompensationen von Messoverhead |
| Trigger | Messung gewünschter Zeiträume |
| Trace-Download | Permanent oder selektiv |

FUNKTIONEN

| | |
|-----------------|--|
| SWF | StopWatch on Function, auch als Focus-Messung |
| SWC | StopWatch on Code segment, auch als Focus-Messung |
| CAF | Code Access Frequency, Zugriffshäufigkeit Code |
| DAF | Data Access Frequency, Zugriffshäufigkeit Daten |
| DAF mit RMC/RMF | DAF mit Restrict Measurement on Code / Function |
| UED | User Event on Data, setzt User Event bei Datenzugriff |
| UED mit RMC/RMF | UED mit Restrict Measurement on Code / Function |
| UEC | User Event on Code access, setzt User Event bei Codezugriff |
| Symbolgruppen | Anlegen von Symbolgruppen |
| Address-Range | Start-Ende Bereich, Code-Adr. Symbol-Name, Source-Code, ... |
| STA | Stopp Trigger Access setzen bei Zugriff auf Code / Daten |
| STA immediate | Direktes Setzen einer Stopp-Trigger-Access Adresse |
| NCA | Next Code Access, Sprungzielermittlung |
| CPC | Code Path Counter, Codezugriffsermittlung einer Adresse |
| Weiteres | Source Viewer, Speichern, dyn. Stopwatches, dyn. Konfiguration |



FUNKTIONALE WILDCARDS

Im Lebenszyklus eines Projekts entstehen die Funktionsgruppen sequenziell. **Mit T1.delay können Nutzer für noch fehlende Funktionen die CPU mit sog. Delays füllen,** die anstelle des Funktionscodes die CPU belasten. So kann ein System bereits ohne funktionalen Code zeitlich abgeschätzt werden. Besonders in großen Projektgruppen und / oder in der Zusammenarbeit mit Zulieferern wird T1.delay zum unverzichtbaren Planungsbaustein.

FUNKTIONEN

| | |
|-----------------------------|---|
| Verzögerungsfunktion | Erzeugt künstliche Verzögerung und repräsentiert Platzhalter für später folgenden funktionalen Code |
| Auto-Inkrement | Verzögerung kann automat. inkrementiert werden. So können empirisch Grenzwerte ermittelt werden. |
| Reset DLY | Reset DeLaY, Rücksetzen der Verzögerung |

T1 test

VERIFIKATION AUF SYSTEM- UND CODEEBENE

T1.test bietet vollständig automatisierte Laufzeitmessungen. Die Anbindung zur Testumgebung erfolgt durch XML basierter Skripte mittels der .NET, .EXE oder Python Schnittstelle. Besonders effizient ist die Integration der Laufzeitverifikation in funktionale Tests, (z.B. HIL). Auf diesem Wege werden beide Disziplinen – Funktion und Laufzeitverhalten – gleichzeitig verifiziert. T1 generiert die zugehörige Dokumentation der Laufzeitanalyse.

FUNKTIONEN

| | |
|------------------------|--|
| Start T1 | Startet T1 von externem Tool |
| Start T1 Script | Startet ein T1-Script von einem externen Tool |
| Stop T1 | Beendet T1 durch ein externes Tool |
| Auto-Log | Erstellt automatisch ein Logfile der Automatisierung (Dokumentation) |

Übersicht automatisierter Schlüsselbefehle (Auszug):

Verbindungsaufl./abbau, Start/Stop Messung, Auto-Download (permanent, wait on trigger, beenden), HTML-/CSV-Report generieren, Projekt laden, Trace laden/ sichern, Delays konfigurieren / aktualisieren, Reset / Start / Stopp Delay-Inkrement., Verzeichnisse festlegen (für Laden / Speichern), Constraint-Check, Application-Maske setzen, T1.flex Messungen durchführen (SWF, SWC, CAF oder DAF), Messen von Symbolgruppen mit / ohne Fokus.

T1 cont

KONTINUIERLICHE ON-TARGET LAUFZEITANALYSE

Anders als bei T1.scope erfolgt bei T1.cont die **quantitative Auswertung bereits auf dem Target** - ein ideales Verfahren für Langzeitmessungen. Alle gewünschten Tasks und Interrupts werden einbezogen. Minimal- und Maximalwerte lassen sich lückenlos erfassen - mittels des „focussed Measurements“ auch durchschnittliche Werte. Optionale „Constraints“ erlauben es, frühzeitig auf die Verletzung frei definierbarer Grenzen aufmerksam zu machen. Dies kann mittels Trigger und T1.scope Visualisierung und/oder per Callback erfolgen. Mit Callback kann die Anwendung eventuell erforderliche Maßnahmen ergreifen, wie beispielsweise die Eintragung eines Fehlers in den Fehlerspeicher. Somit können Laufzeitprobleme auch bei großen Erprobungen mit minimalem Aufwand entdeckt werden.

FUNKTIONEN

| | |
|-----------------------------|---|
| Messwertermittlung | CET, GET, IPT, RT, DT, ST, CPU-Last |
| Stopwatches | Lückenlos Min./Max CET/GET auch dynamische T1.flex Stopwatches |
| Dyn. Konfigurationen | Ändern zur Laufzeit der Berechnungsgrößen, Ergebnisgrößen, überwachte Constraints |
| Anzeige | Grafische oder tabellarische Darstellung |
| Speichern / Laden | T1.cont Trace Daten für Weiterverarbeitung |
| Fokus-Messung | Fokus auf spez. Timinggrößen und definierte Zeiträume |
| Fokus-Gruppen | Fokus-Messgruppen von Elementen, wie alle Tasks, ISRs, etc. |
| Reset | Reset von Ergebnissen |
| Constraint On/Off | Gezieltes Ein/Aus der Constraint -Überwachung |
| Analyse On/Off | Gezieltes Ein/Aus der Event-Analyse und Berechnungen |
| live-Status | Abfrage des Status mit der GUI |

PORTIERUNG / SCHNITTSTELLEN

CONTROLLER

| Hersteller | Core | Controller | Anmerkungen |
|--------------|----------------------|----------------------|------------------------------|
| Atmel | AVR MEGA | ATmega32, ... | T1.flex nicht verfügbar |
| Atmel | AVR XMEGA | ATxmega32, ... | T1.flex nicht verfügbar |
| Freescale | CPU12/CPU12X | HC12/S12/S12X | T1.flex noch nicht verfügbar |
| Freescale | e200 | MPC5xxx | volle Unterstützung |
| Freescale | Power Architektur | MPC5xx | volle Unterstützung |
| Fujitsu | F ² MC-16 | F ² MC-16 | T1.flex nicht verfügbar |
| Infineon | TriCore 1.3.1 | TC1797, ... | volle Unterstützung |
| Infineon | TriCore 1.6 | TC1798, ... | volle Unterstützung |
| Infineon | TriCore | MultiCore | volle Unterstützung |
| Infineon | C166 v1/2 | (X)C16x / ST10 | T1.flex noch nicht verfügbar |
| Microchip | dsPIC | dsPIC33FJ, ... | T1.flex nicht verfügbar |
| NXP | Cortex M0 | LPC11xx | T1.flex nicht verfügbar |
| NXP | Cortex M3 | LPC17xx | volle Unterstützung |
| Renesas | V850e1 / e2 (dual) | V850, PX4,... | volle Unterstützung |
| Texas Instr. | Cortex M3 | TMS570 | volle Unterstützung |
| Texas Instr. | Cortex R4 | TMS570 | volle Unterstützung |

RTOS

| Hersteller | Name | Hersteller | Name |
|------------|----------|--------------|------------------|
| GLIWA GmbH | gliwOS | Elektrobit | ProOSEK, OEKtime |
| ETAS | ERCOSEK | Elektrobit | tresOS |
| vector | osCAN | ETAS | RTA-OSEK |
| Delphi | PharOS | KPIT Cummins | KPIT |
| Micrium | µC/OS-II | Kunden-OS | diverse |

COMPILER

| Hersteller | Compiler | Core |
|----------------------|--------------|---------------------------------------|
| Altium | Tasking | C166 und TriCore |
| Atmel | AVR (X)MEGA | AVR (X)MEGA |
| Freescale | Metrowerks | CPU12(X) und e200 |
| Fujitsu | Softune | F ² M-16 |
| GNU | gcc | TriCore, (X)MEGA, Cortex M0/M3, dsPIC |
| Green Hills Software | GHS Compiler | e200 (dual), V850e1/e2 |
| Texas Instruments | TI Compiler | Cortex M3, Cortex R4 |
| Wind River Systems | Diab Data | Power Architektur und e200 |

FILE-FORMATE IMPORT / EXPORT

| Type | Erklärungen |
|-------------|--|
| binary | Timing- und Trace-Daten |
| txt | Trace-Daten |
| csv | Trace- und Timing-Daten |
| html | Timingdaten, Systemkonfiguration |
| ATF | ALL-TIMES Timingdaten (basierend auf EU-ALL-TIMES Projekt) |
| CTF | Common Timing Format, Plattform übergreifendes Format |
| OT1 | Open Timing Format |
| elf | ELF-Formate |
| XTC | Schnittstelle zur Absint Toolsuite aiT |
| SymTA/S XML | Schnittstelle zur Syntavision Toolsuite SymTA/S |
| INCHRON XML | Schnittstelle zur INCHRON Toolsuite |
| T1 script | Automatisierungsskripte für T1 |
| Artop | Schnittstelle zu ARTIME |

Die GLIWA GmbH besitzt jahrelange
Erfahrung in der Optimierung von Laufzeitverhalten.
Viele Kunden profitieren bereits von den Coaching-Aktivitäten.
Dabei wird wertvolles Know-How
in die Unternehmen gebracht!

WIR ENTWICKELN PRÄZISION



www.gliwa.com