

Wirkketten-Analyse für gemischte Systeme aus AUTOSAR Classic und Adaptive Plattformen

Olaf Schmidt, Dr. Ralf Münzenberger, Philip Rehkop, INCHRON GmbH; Armin Stingl, iSYSTEM GmbH; Michael Reiß, Elektrobot GmbH

Für hoch automatisiertes und autonomes Fahren wird eine hohe Rechenleistung benötigt, so dass High Performance Controller, die auf der AUTOSAR Adaptive Plattform basieren, mit Steuergeräten, die auf der AUTOSAR Classic Plattform basieren, integriert werden müssen. Dabei ist es wichtig von Anfang an im Projekt Ende-zu-Ende-Wirkketten, die sich über alle Steuergeräte erstrecken, zu designen und zu testen.

Dieser Beitrag zeigt umfassend die Symbiose aus AUTOSAR Classic und Adaptive Plattformen, aus statischer und dynamischer Architektur und aus Simulation und Messung am Target. Er entstand aus einer Kooperation der Firmen Elektrobot, iSYSTEM und INCHRON.

1 Motivation

Die Anforderungen an die E/E Architektur im Fahrzeug haben sich durch die derzeitigen Trends, wie z.B. das automatisierte Fahren oder die Elektromobilität, verändert. Vor allem das automatisierte Fahren verlangt immer leistungsfähigere Steuergeräte (electronic control unit – ECU), welche die Informationen aus den verschiedenen Eingangsquellen z.B. zur Berechnung einer automatisierten Fahrreaktion nutzen. Das berechnete Umfeldmodell des Fahrzeugs soll außerdem von anderen Fahrfunktionen genutzt werden können.

Die genannten Anforderungen können in einer klassischen E/E Domain Architektur nur schwer umgesetzt werden. Stattdessen wird auf eine zentralisierte E/E Architektur gesetzt, in der die verschiedenen smarten Sensoren, Aktuatoren und weiteren ECUs im Fahrzeug mit nur wenigen, aber sehr leistungsfähigen Steuergeräten (high-performance controller) verbunden sind. Dies gewährleistet eine hohe Flexibilität und erlaubt, dass Fahrzeugfunktionen dynamisch auf den leistungsfähigen Steuergeräten verteilt werden können.

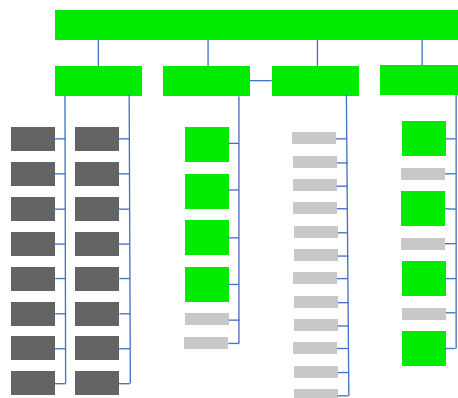


Abbildung 1: Domain Architektur

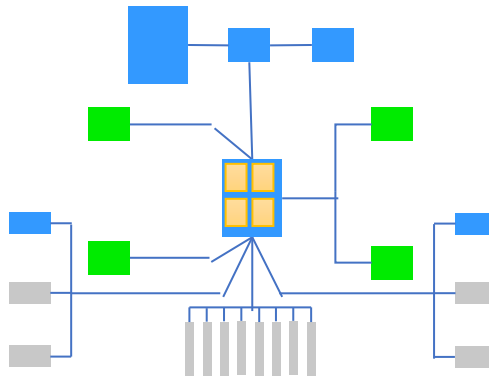


Abbildung 2: Zentralisierte Architektur

Bei einer zentralisierten E/E-Architektur im Fahrzeug wird es eine große Anzahl von Wirkketten zw. High Performance Controllern und den „klassischen“ Steuergeräten geben. Die Gesamtfunktionalität ergibt sich aus dem Zusammenspiel beider Plattformen. Dabei müssen die Echtzeitanforderungen an das Gesamtsystem und Wirkketten entlang der Datenflüsse vom Sensor bis zum Aktuator berücksichtigt werden, inkl. serviceorientierter Kommunikation.

Eine Simulation wird zur Absicherung des Timingverhaltens der Komponenten entlang der Wirkketten eingesetzt. Datenfluss-Wirkketten repräsentieren dabei Daten, die entlang einer Prozesskette betrachtet werden. Diese haben neben dem Inhalt auch oft zeitliche Anforderungen wie Latenz, Prozessreihenfolge und Datenalter.

Im Zuge der Implementierung werden Traces im laufenden Steuergerät aufgenommen. Die Timing-Anforderungen werden in der Simulation und den Traces (Implementierung) geprüft. So können Anforderungen frühzeitig designed und geprüft werden und später die Implementierung verifiziert werden.

Dieses Paper beschreibt den Ablauf anhand eines Beispiels, bei dem ein Sensor auf einer ECU mit Classic Plattform implementiert ist und über Ethernet eine Verbindung zur einer ECU mit Adaptive Plattform hergestellt wird. Die Wirkketten werden vorab entworfen, simuliert und getestet und dann die Anforderungen mit der Implementierung nochmal getestet. Dieses Zusammenspiel erfordert neue Tools und Integrationen auf Seite der Steuergerätesoftware, Tracing Tools und Simulation & Analyse. In der Zusammenarbeit von Elektrobit, iSYSTEM und INCHRON ist eine integrierte Lösung für Kunden entstanden, die die Classic und Adaptive Plattformen überspannen.

2 AUTOSAR Classic vs. Adaptive Plattform

Die Steuergerätearchitektur basierte bisher hauptsächlich auf der AUTOSAR Classic Plattform. Die AUTOSAR Classic Plattform nutzt ein statisches Betriebssystem, welches auf dem OSEK Betriebssystem (Offene Systeme und deren Schnittstellen für die Elektronik im Kraftfahrzeug) basiert und von AUTOSAR um einige Funktionalitäten erweitert wurde. Einerseits ist ein solch statischer Ansatz sehr hilfreich für die Entwicklung sicherheitskritischer Funktionalitäten bzw. ausreichend für smarte Sensor / Aktuator Steuergeräte, andererseits ist die gewünschte Flexibilität damit nicht zu erreichen. Aus diesem Grund wurde die AUTOSAR Classic Plattform

um die AUTOSAR Adaptive Plattform ergänzt. Im Unterschied zur Classic Plattform spezifiziert AUTOSAR in der Adaptive Plattform kein eigenes Betriebssystem. Stattdessen wird die Verwendung eines POSIX-Betriebssystems (POSIX Profile PSE51) gefordert. Somit steht auch Linux, welches im Automotive Bereich z.B. bei Infotainment Systemen bereits eingesetzt wird, als Betriebssystem für die Adaptive Plattform zur Verfügung.

Zusammenfassend lässt sich aus den verschiedenen Anforderungen schlussfolgern, dass in einer zentralisierten E/E Architektur beide AUTOSAR Plattformen eingesetzt werden. Somit stellt sich natürlich die Frage, in welcher Art und Weise die Daten zwischen der Classic und der Adaptive Plattform ausgetauscht werden können. Sowohl die Classic als auch die Adaptive Plattform unterstützt das Prinzip der serviceorientierten Kommunikation über Ethernet und löst damit die bisherige signalbasierte Kommunikation ab, die auf den Automotive Kommunikationsbussen CAN, LIN und FlexRay genutzt wird. Die Daten zwischen Steuergeräten werden nur noch dann ausgetauscht, wenn die von einem Steuergerät angebotenen Services von anderen Steuergeräten benötigt und konsumiert werden. Als Protokoll kommt hierbei SOME/IP (Scalable service-Oriented MiddlewarE over IP) zum Einsatz.

3 Demonstrationsprojekt

Das von Elektrobit, INCHRON und iSYSTEM aufgesetzte Demonstrationsprojekt soll diesen Aspekten Rechnung tragen.

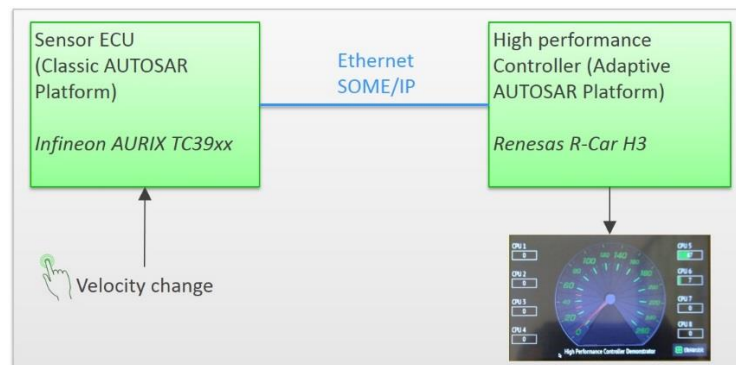


Abbildung 3: Demonstrationsprojekt

Eine Sensor ECU, auf welcher die Classic Plattform Software von Elektrobit genutzt wird (EB tresos), liest einen Taster aus und leitet daraus ein Geschwindigkeitssignal ab. Der Service „Geschwindigkeitssignal“ wird über die sogenannte „Service Discovery“ anderen Steuergeräten angeboten. Als Hardwareplattform kommt ein Infineon AURIX TC39x zum Einsatz.

Eine zweite ECU – der High-performance Controller – abonniert den Service „Geschwindigkeitssignal“ und empfängt danach dieses Signal. Zur Übertragung wird das SOME/IP Protokoll genutzt. Die aktuelle Geschwindigkeit wird auf einem Tachometer dargestellt. Die High-performance Controller Software basiert auf der Adaptive Plattform Lösung und der Linux Distribution von Elektrobit (EB corbos). Als Hardwareplattform wird ein Renesas R-Car H3 genutzt.

4 Wirkketten und Architekturebenen

Datenfluss-Wirkketten sollten im gesamten Fahrzeug- und Steuergeräte-Entwicklungszyklus auf durchgängig auf allen Architekturebenen betrachtet werden. Zu Beginn werden auf der Kundenfunktionsebene Sensoren und Aktuatoren für die Kundenfunktion miteinander verbunden und die Latenzzeiten festgelegt.

Auf der Funktionalen Ebene werden die Kundenfunktionen in Blöcke zerlegt, die Teilfunktionen nacheinander oder parallel abarbeiten. Jede von diesen Funktionen bekommt ein Zeitbudget entlang der Wirkkette zugeteilt und kann in der Simulation getestet werden.

Die Funktionen werden auf der OEM Systemebene auf verschiedene Steuergeräte verteilt. Die Steuergeräte sind über verschiedene Busse miteinander verbunden. Die Zeitbudgets müssen entsprechend entlang der Wirkketten auf die Steuergeräte und Kommunikationszeiten verteilt werden.

Im ECU System bewegt sich der Fokus zu einem einzelnen Steuergerät und den Funktionen und Wirkketten im Steuergerät. Hier werden verschiedenen CPU, SoC, Cores und ECU-interne Busse betrachtet.

In der Ebene der Softwarearchitektur wird festgelegt, wie die Funktionen/Runnables auf Tasks und ISRs verteilt werden und die Parameter (Corezuordnung, Priorität, usw.) dieser Elemente werden festgelegt. Da in der Simulation das Scheduling berücksichtigt wird, müssen aus den zuvor festgelegten Zeitbudgets Nettoausführungszeiten abgeleitet werden.

Auf jeder Ebene kann durch Simulation und Test die Wirkketten und Anforderungen an das Gesamtsystem geprüft werden. Durch virtuelle Verifikation können in der Implementierungs- und Testphase Testfälle abgeleitet werden, die an einem realen Steuergerät nicht so durchführbar wären.

In dem Demonstrationsprojekt wurden Wirkketten auf OEM System, ECU System und Software in einem chronSIM Modell kombiniert und simuliert:

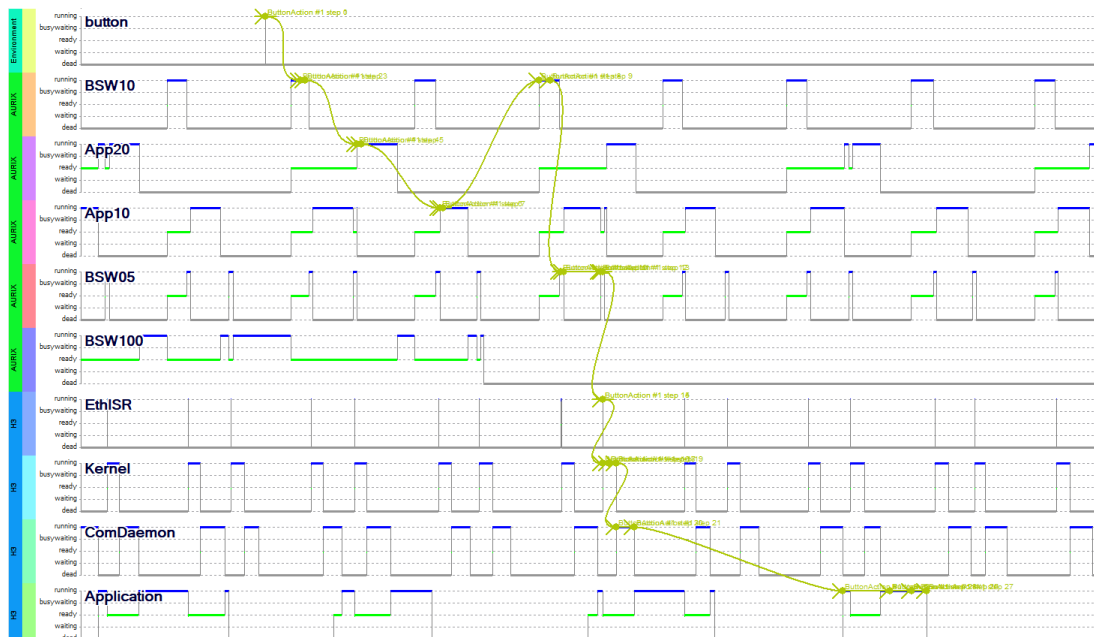


Abbildung 4: Simulation der Wirkkette mit INCHRON chronSIM

5 Synchrone Messung parallel laufender Prozessoren (iSYSTEM)

5.1 Messaufbau

Das zeitliche Verhalten der Wirkkette wird bei beiden Prozessoren mithilfe der on-chip Trace Logik untersucht.

Beim Infineon TC39x wird hierzu das Multi-Core Debug System (MCDS) benutzt. Als Schnittstelle zwischen Prozessor und Trace Tool wird das DAP Interface gewählt.

Der Renesas R-Car H3 Prozessor implementiert die CoreSight Components von ARM. Jeder Core verfügt über eine Embedded Trace Macrocell (ETM), wobei die ETM der A5x Cores nur Instruction Trace unterstützt, jedoch kein Data Trace. Zusätzlich bietet der R-Car noch ein System Trace Module (STM). Als Trace Interface dient hier der ARM High-Speed Serial Trace Port (HSSTP).

Um ein zeitsynchrones Tracing beider Prozessoren zu erreichen, sind die iSYSTEM iC5700-internen Zeitstempel zueinander synchronisiert. Ein optionaler Sync-Port erlaubt es, zwei iC5700 miteinander zu koppeln.

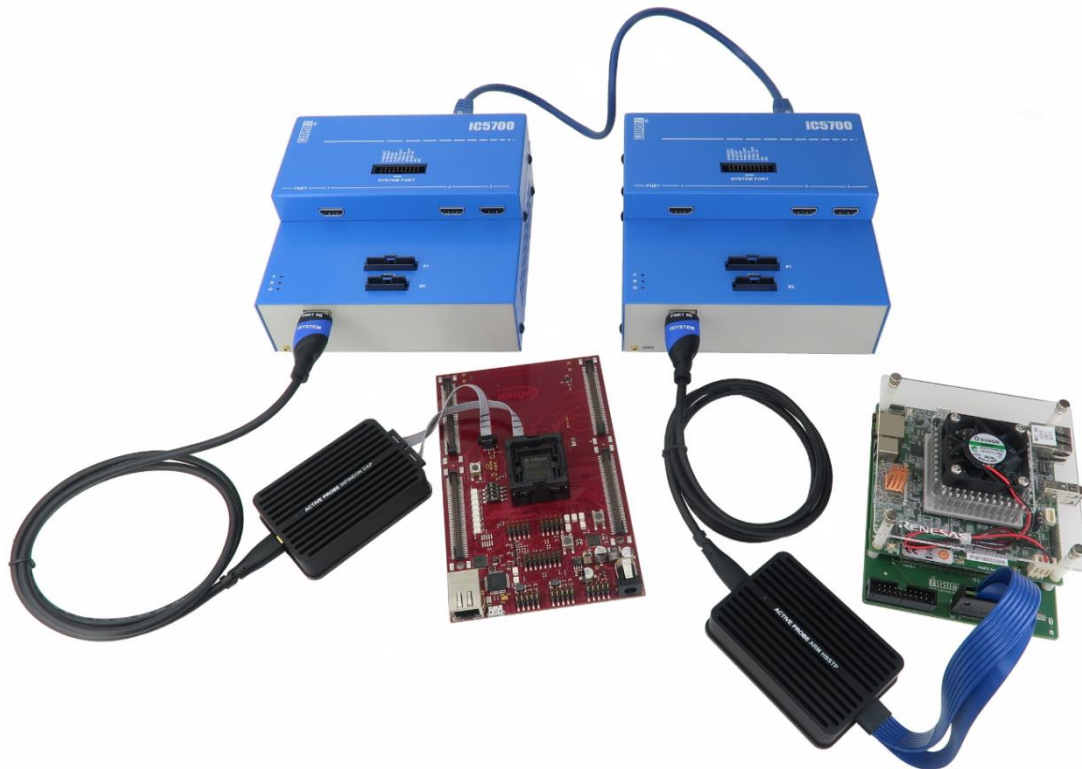


Abbildung 5: Messaufbau bestehend aus zwei zeitsynchronisierten iSYSTEM iC5700 On-Chip Analyzer

5.2 Effizientes Tracing mithilfe von Code Instrumentierung

Auf beiden Prozessoren spielt die verfügbare Trace Bandbreite eine entscheidende Rolle. Die naheliegendste Trace-Methode zur Event-Chain Analyse scheint ein Function Profiling basierend auf Instruction Trace zu sein. Eine effizientere Alternative hierzu ist es nur die wirklich relevanten Ereignisse zu signalisieren. Das heißt, der Trace generiert nur Daten zur zeitlichen Markierung relevanter Events, wie

etwa Entry und Exits relevanter Funktionen innerhalb der Wirkkette oder OS Context Switches.

Auf der Classic Plattform ECU ist ein instrumentierungs-basiertes OS Task und ISR2 Tracing recht komfortabel realisierbar, da das OS die entsprechenden Trace Hooks bereits implementiert. Diese Hook Makros müssen nur durch Trace Tool spezifischen Code ersetzt werden. Für das instrumentierungs-basierte Tracing von Runnables werden die Virtual Functional Bus (VFB) Tracing Hooks der RTE genutzt.

Auf der Adaptive Plattform ECU wird das instrumentierungs-basierte Tracing von OS Kernel Events, Adaptive Komponenten wie etwa ARA:COM, sowie der Applikation dadurch realisiert, dass direkter Schreibzugriff auf den STM Stimulus Port erlaubt wird. Dies wird mithilfe einer minimal-invasiven Code Instrumentierung des Linux Kernels erreicht. Sowohl OS Prozess/Thread Zustände wie auch die Ausführung von Funktionen innerhalb der Wirkkette werden über dedizierte STM Channels signalisiert.

5.3 Konfiguration der Messtools

Trace Konfiguration

Beim TC39x der Classic Plattform ECU schreibt der Trace Instrumentierungscode die relevanten Informationen, also Task IDs, ISR2 IDs und Runnable IDs in drei speziell dafür allokierte globale Variablen, die von der Data Trace Unit des MCDS observiert werden und über das DAP Interface zum Trace Tool kommuniziert werden.

Auf dem R-Car kommt die System Trace Macrocell (STM) zum Einsatz. Dazu werden die Daten, wie etwas OS Process/Thread Handles oder Function IDs auf einen sogenannten Stimulus Port des STM geschrieben. Solch ein Schreibzugriff initiiert eine STM Trace Message, die aus STM Channel ID, Data Payload und Timestamp besteht. Die STM Trace Messages werden über HSSTP zum Trace Tool geschickt. Der Stimulus Port des STM ist in Channels unterteilt. Die Channel ID kann dazu verwendet werden, verschiedenartige Events zum Trace Tool zu signalisieren. So können z.B. Function IDs über Channel 2 signalisiert werden während eine OS Thread ID Signalisierung den Channel 1 nutzt.

Profiler Konfiguration

Beim Profiling werden die aufgezeichneten Trace Daten interpretiert. Der kritische Aspekt hierbei ist die Definition der Interpretationsregeln. Im Falle des iSYSTEM Profilers kann dies durch eine XML Beschreibung geschehen. Es können Profiler Objekte definiert werden, um z.B. dem Profiler mitzuteilen, dass die ID der ausgeführten Funktion über den STM Channel 1 signalisiert wird.

Abbildung 6 zeigt eine iSYSTEM Profiler Timeline Darstellung der Wirkkette innerhalb der AUTOSAR Adaptive ECU, inklusive relevanter OS Threads und Funktionen.

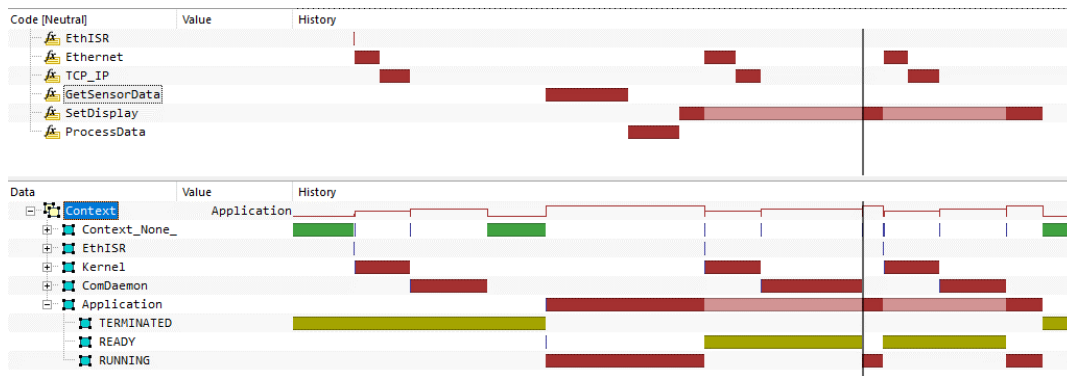


Abbildung 6: iSYSTEM Profiler Timeline Darstellung der Wirkkette innerhalb der AUTOSAR Adaptive ECU

Das Ergebnis der Trace/Profiler Analyse kann z.B. im sogenannten Best-Trace-Format (BTF) exportiert werden. Dieses BTF Format kann von vielen Timing Analyse/Modelling Tools wie z.B. der INCHRON Toolsuite eingelesen werden.

6 Test der Wirkketten

Die Traces werden von INCHRON chronVIEW auf Basis der Zeitanforderungen getestet. Darüber hinaus können mithilfe der Simulation virtuelle Tests umgesetzt werden indem Situationen mit Hilfe von Stochastik oder What-if-Analysen eingestellt werden, die so am realen Target nicht mit sinnvollem Zeitaufwand testbar wären. Dadurch wird eine erhöhte Testtiefe erreicht.

7 Zusammenfassung

Das Zusammenspiel von AUTOSAR Classic und Adaptive Plattformen wird die Architektur der zukünftigen Fahrzeuge bestimmen.

Dabei ist zum einen wichtig, dass die eingesetzten Softwarekomponenten und Werkzeuge optimal aufeinander abgestimmt sind. Das umfasst die Bereiche Architektur/Design, ECU Basissoftware und Messwerkzeuge.

Zum anderen ist der Fokus auf das Design von Wirkketten schon in der frühen Phase der Entwicklung ein Garant für die Vermeidung von Timing Problemen.

Literaturverzeichnis

[1] Thomas Bock, Dr. Henning Kleinwechter (Volkswagen), Dr. Ralph Sasse (OpenSynergy), Armin Stingl (iSystem), Dr. Ralf Münzenberger, Philip Rehkop, Olaf Schmidt (INCHRON): Einsatz von Virtualisierung für sichere Softwarearchitekturen, ESE Kongress 2017

Autoren



Olaf Schmidt, INCHRON GmbH, Business Development,
olaf.schmidt@inchron.com



Philip Rehkop, INCHRON GmbH, Professional Services Engineer, philip.rehkop@inchron.com



Dr. Ralf Münzenberger, INCHRON GmbH, CEO and Mitgründer, ralf.muenzenberger@inchron.com



Armin Stingl, iSYSTEM AG, System Ingenieur, armin.stingl@isystem.com



Michael Reiß, Elektrobit Automotive GmbH, michael.reiss@elekrobit.com