

R-IN Consortium Forum 2018

京都マイクロコンピュータ株式会社

ソフトウェアの信頼性向上を支援する、
RTOS統合開発プラットフォームSOLIDのご紹介

バグの自動検出も 可能な新開発環境

SOLID

京都マイクロコンピュータについて

- 設立 1985年
- CEO 山本 彰一
- 国内拠点
 - 本社/開発拠点 京都市西京区
 - 東京オフィス 港区新橋
- 海外販売
 - 韓国代理店 Nextech Solutions Co., Ltd
 - インド代理店 Celestial Systems Pvt. Limited
- 主要製品
 - JTAGエミュレータ PARTNER-Jet2
 - GCC コンパイラ exeGCC
 - 開発プラットフォーム SOLID



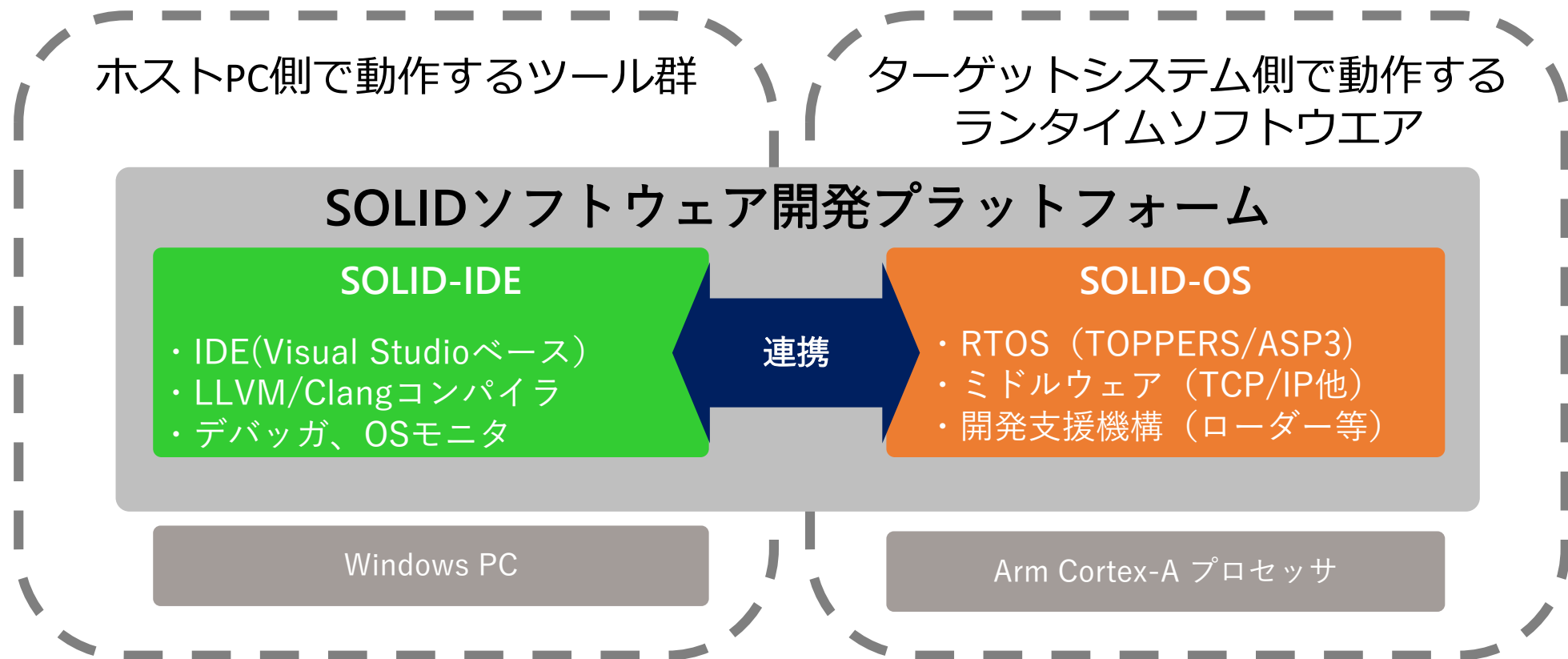
SOLID とは

- 開発環境とRTOSが一体化提供される、組み込み向けソフトウェア開発プラットフォーム
- SOLID-IDE
 - VisualStudioベースのIDE
 - GCC/Clangコンパイラベースのツールチェーン
 - KMCが創業以来培ってきたデバッガ
- SOLID-OS
 - リアルタイムOS（TOPPERSカーネル）
 - Armプロセッサに対応した専用ランタイムライブラリ
- これを一つ用意するだけで、開発がスタートできる「優れモノ」

バグの自動検出も 可能な新開発環境

SOLID

SOLIDの構成



バグの自動検出も
可能な新開発環境

SOLID

同じようなものは、今までもあった
のではないか？

何が新しく、何が違うのか？

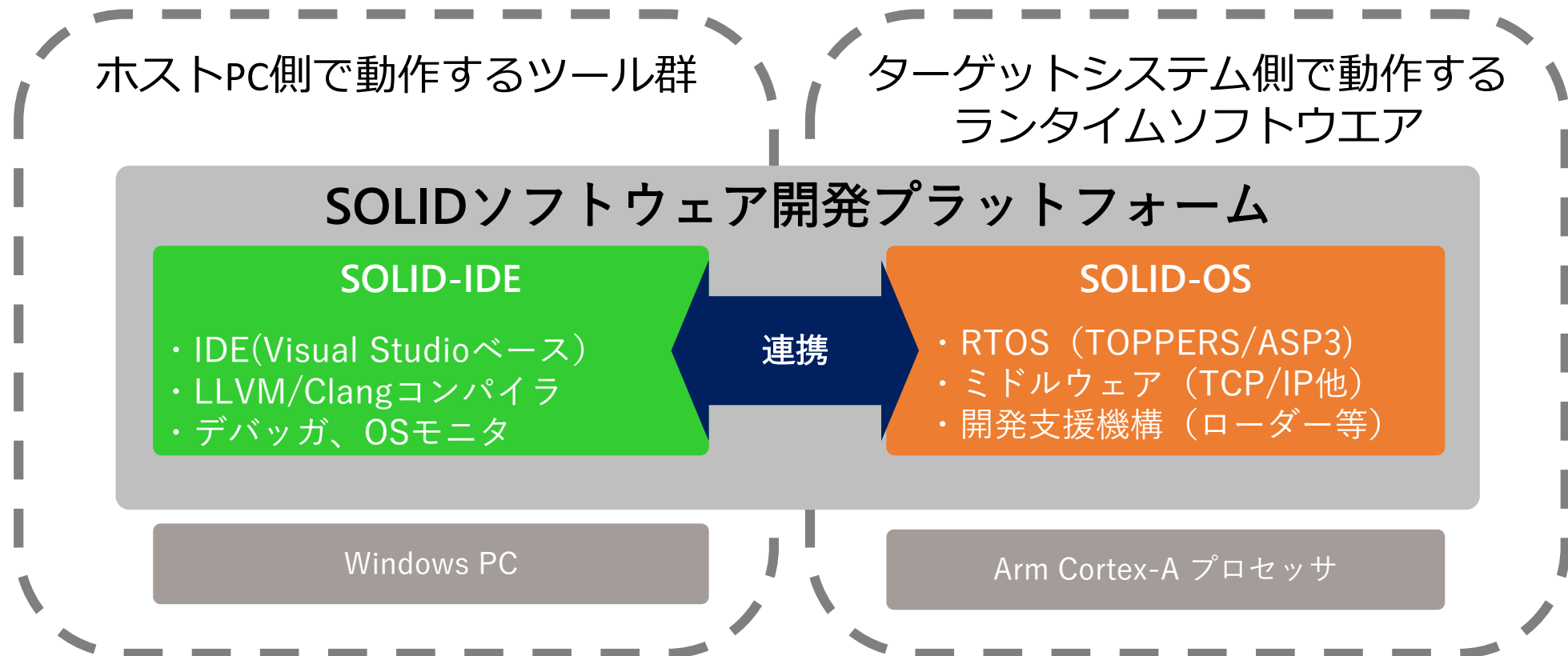
バグの自動検出も 可能な新開発環境

SOLID

ここが新しい！

SOLIDの構成

ホストPCとターゲット、双方の**ソフトウェアの専用設計**が特徴です



バグの自動検出も
可能な新開発環境

SOLID

連携というより、むしろ、各々のツールが専用設計されている

- RTOS実装とコンパイラの専用化
- RTOS実装とデバッガの専用化
- コンパイラとデバッガの専用化
- IDEを、RTOS,コンパイラ,デバッガに合わせて専用化

とても排他的だが、その分、使う時には一体化によるメリットが大きく享受される

なぜSOLIDを？

enjoy **D**evelopment

デバッグツールベンダであるKMCが、
なぜいまさらRTOSベースの開発環境を
提供するのか？

バグの自動検出も
可能な新開発環境

SOLID

組み込みアプリケーションソフトウェア開発の普遍的な課題

- ・ 納期どおりに開発する
- ・ 品質を確保する
- ・ 快適な操作性
- ・ 出来るだけ安価に

これらを実現するための開発環境として生まれたのが、SOLID
開発プラットフォームです。

バグの自動検出も
可能な新開発環境

SOLID

開発ツールベンダーである京都マイクロコンピュータの着目点

- ITRON開発環境への新提案という切り口で、
- ITRON含むランタイムと開発環境を専用設計
- 構成要素は コンパイラ、ITRON+ランタイム、IDE、デバッガ

いわゆる、RTOSベースの統合環境が誕生しました。

「これを一つ用意するだけで、開発がスタートできる！」

バグの自動検出も
可能な新開発環境

SOLID

SOLIDを使うと。。。。

- ・立ち上げ簡単で、開発の開始コストを低減
- ・自動バグ検出で、デバッグ・テストコストを低減
- ・チーム開発の悩みを解決できる!!

立ち上げ簡単で、 開発開始コストを低減!!

ツールとRTOSがセットで検証確認不要。

RTOSも数個のAPI実装で動作可能。

専用のArm Cortex用ライブラリと連携出来るGUI設定。

バグの自動検出も
可能な新開発環境

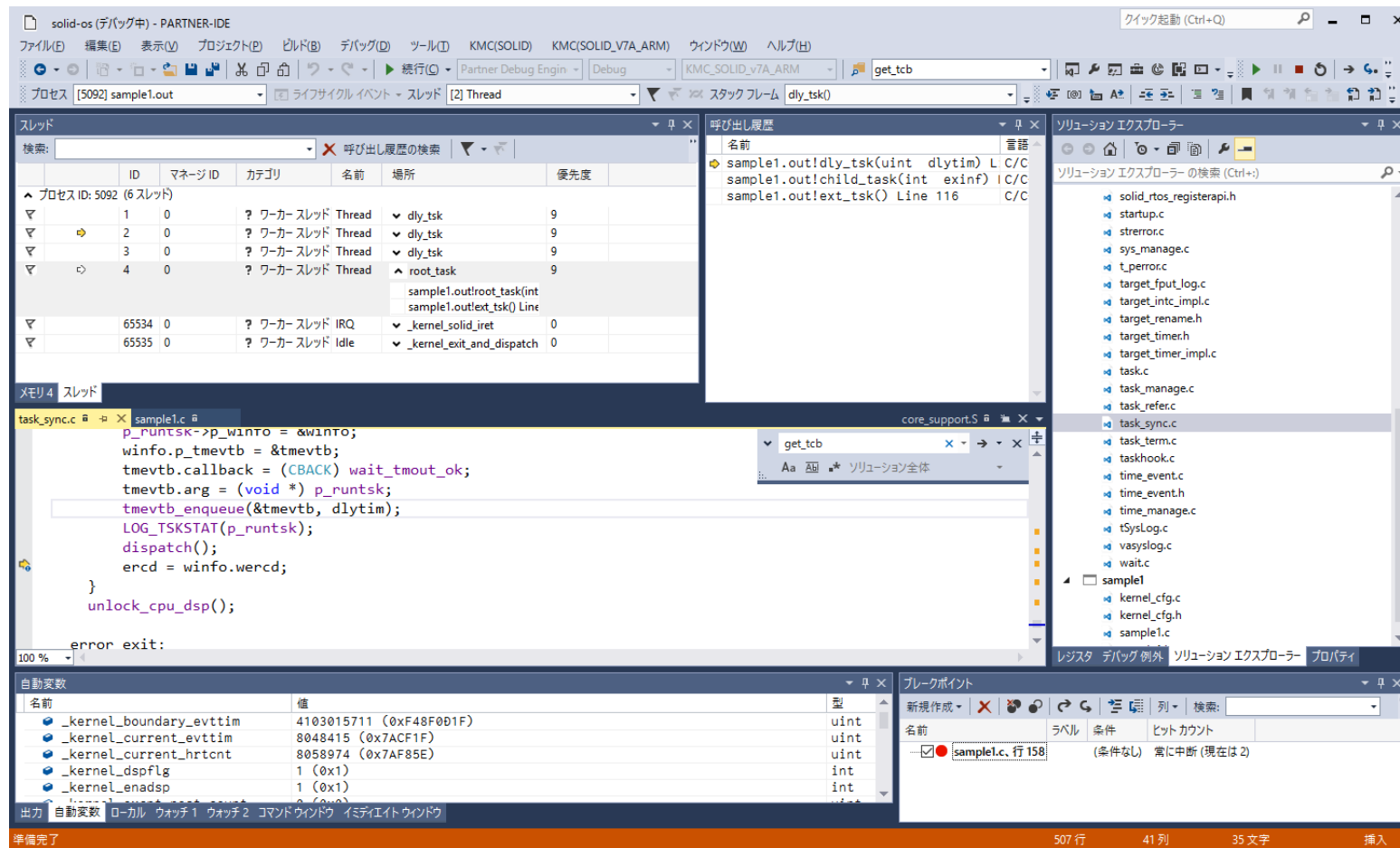
SOLID

- **使いやすくて簡単**なVisualStudio
- RTOSの**立ち上げがきわめて簡単**
 - ITRONカーネル（TOPPERS/ASP3）を、デバイスやボード依存を排除した状態で供給
 - 数個（タイマと割り込み関係）の専用APIを実装するだけで、簡単にRTOSが動作
 - RTOSのランタイムロイヤリティは発生しません（SOLID開発ツールの利用料のみ）

バグの自動検出も 可能な新開発環境

SOLID

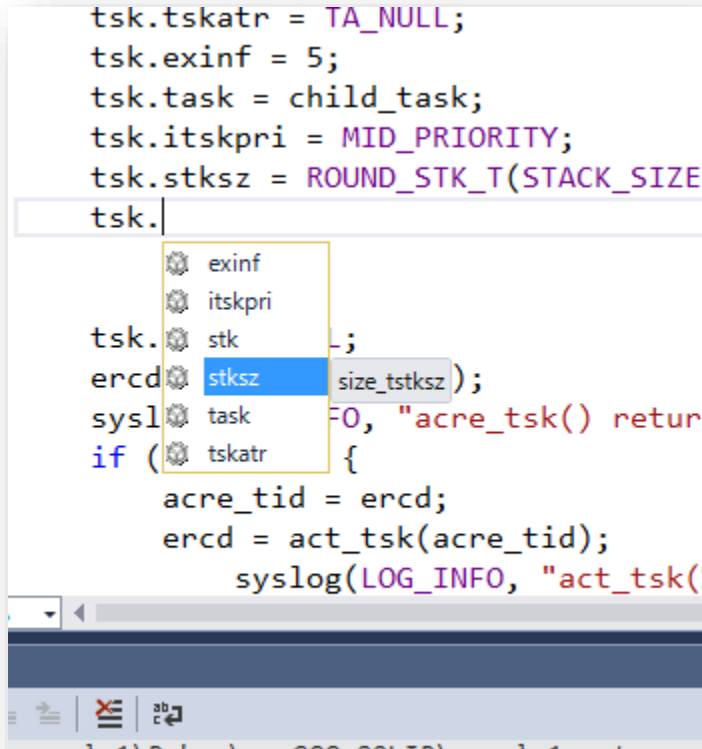
Visual Studio によるIDE



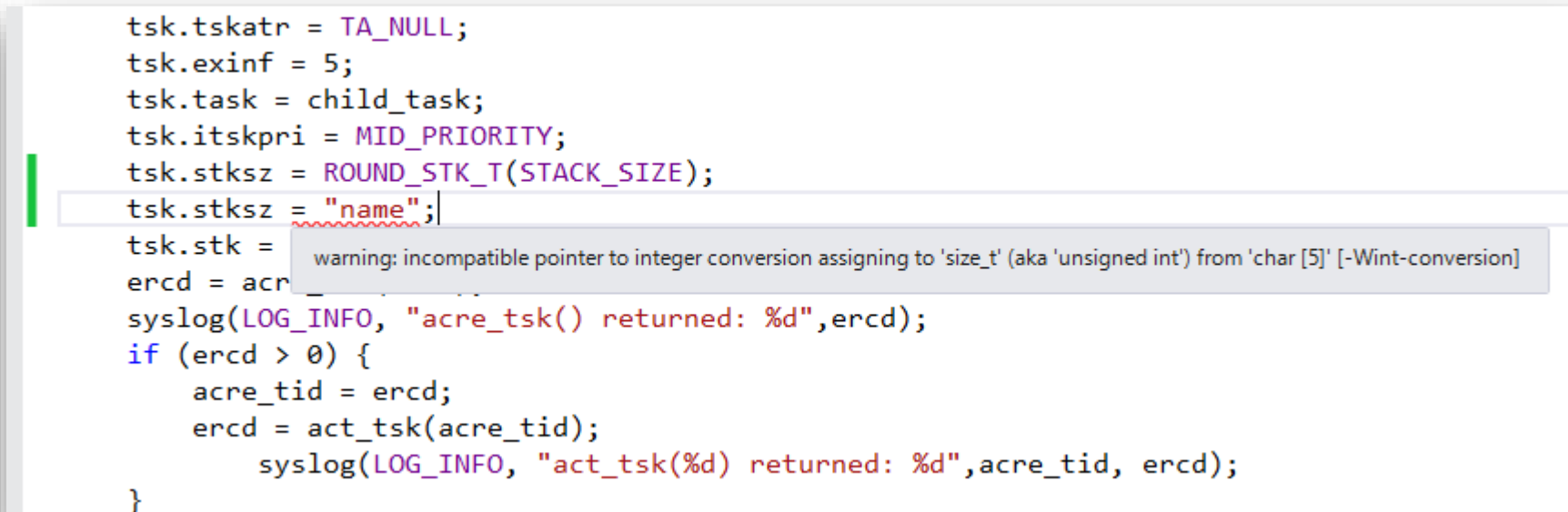
バグの自動検出も 可能な新開発環境

SOLID

確実なコーディングで、build ↔ editの手戻りを少なくします



インテリセンス機能



入力時にリアルタイムにコンパイルエラー・警告を指摘
(バックグラウンドでコンパイラが動作)

バグの自動検出も 可能な新開発環境

SOLID

- ツールチェーンを使いこなす設定や実装も
あらかじめ**準備済み**
 - リンカスクリプトと実行ライブラリが連携するので、
何もしなくても実行命令領域を”書き換え不可”などに設定
- Arm Cortex-AのMMUを、**GUIで設定するだけ**で高機能に利用
 - これもツールの設定処理と、実行ライブラリの連携

バグの自動検出も 可能な新開発環境

SOLID

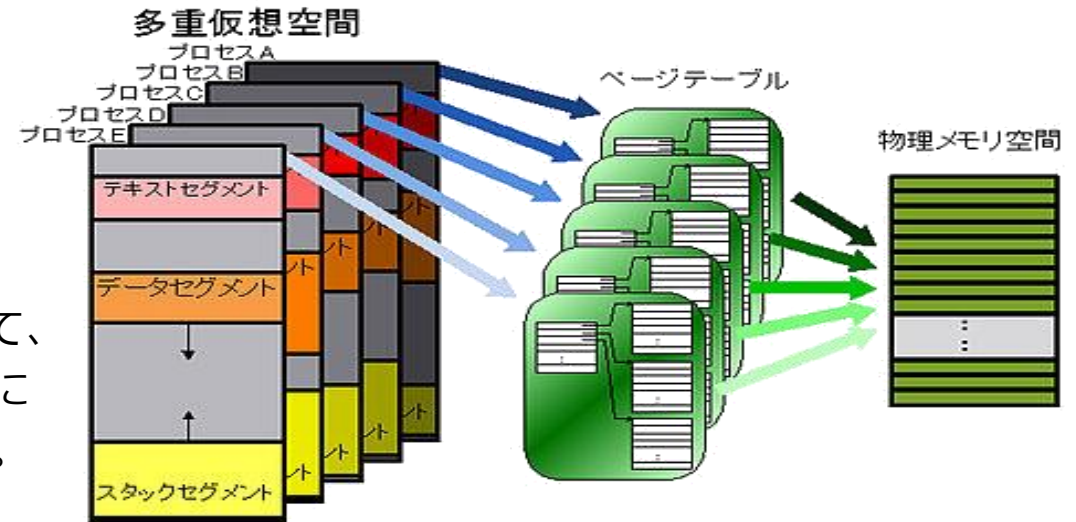
MMUの役割：アドレス変換、メモリ保護、キャッシュの設定を簡単に!!

Arm Cortex-A9などv7A/v8A CPUには、高機能なメモリ管理機構(MMU)が搭載されています。

MMUの主な機能は次のとおりです。

- ・ 仮想アドレスと物理アドレスのアドレス変換機能
- ・ プロセスのための仮想空間を実現するための機能
- ・ 実行モードに応じたメモリ保護

Linux/Androidなどは、この高機能なメモリ管理機能を使って、多重仮想空間のオーバーラップしたプロセスを物理メモリ空間に割り付けて実行する事が特徴のオペレーティングシステムです。



Arm v7A/v8Aプロセッサでは、データキャッシュを有効にするためにはメモリ管理機構を設定（MMU有効化）する必要があります。SHマイコンのように、番地によるキャッシュアクセス／非キャッシュアクセスを区別する方式ではありません。

また、MMU有効下では、CPU命令からのアクセスは常に仮想アドレスになる点にも注意が必要です。

バグの自動検出も 可能な新開発環境

SOLID

OS側ランタイムとツールの専用設計で、MMUを簡単に活用

- Cortex-AプロセッサのMMUを、設定だけで簡単に活用できます
 - CP15レジスタの設定を行うプログラムや、ページテーブルの作成は不要
- 専用設計による次の仕組みで、MMUを簡単に活用できます
 1. IDE上のGUIで、物理アドレスと仮想アドレスのメモリマップを設定
 - アドレスやサイズ、また属性を設定
 2. ビルド時にIDEが、設定内容に基づくMMU設定テーブルを作成
 3. Arm側には上記MMU設定テーブルを元に、MMUを有効化するライブラリが動作
 4. 仮想アドレスやキャッシュ、プロテクションが動作
- MMUが簡単に使えるため、SOLIDはMMUを積極的に活用
 - SOLIDではリンクスクリプトと連動するプロテクション、やスタックフェンス、ELFローダーなど、各種機能がMMUを積極的に利用しています

バグの自動検出も
可能な新開発環境

SOLID

IDEから出来るMMUの設定

memory_map.smm

Address Sanitizer

Add Remove Clear

Name	Attribute	Physical Address	Size	Virtual Address	Description
JPEG_READ_BUF	SOLID_RAM	0x20398000	0x000E8000	0xf6018000	
FRAME	SOLID_IO	0x20480000	0x00180000	0xf6100000	
MEMFS	SOLID_RAM	0x20700000	0x00100000	0xf7000000	
SOLID	SOLID_CORE	0x20800000	0x00200000	0xf8000000	
SLV3	SOLID_IO	0xE8000000	0x00020000	0xF0000000	

Physical Address

Virtual Address

物理アドレスマップ

仮想アドレスマップ

表形式で入力

IDE上で設定すれば、その通りに配置

自動バグ検出で、 デバッグ・テストコストを低減!!

ソースコード解析が標準付属。

先進的な実行時メモリバグ検出機能や、メモリプロテクションで、隠れたバグを簡単にあぶり出し。

バグの自動検出も
可能な新開発環境

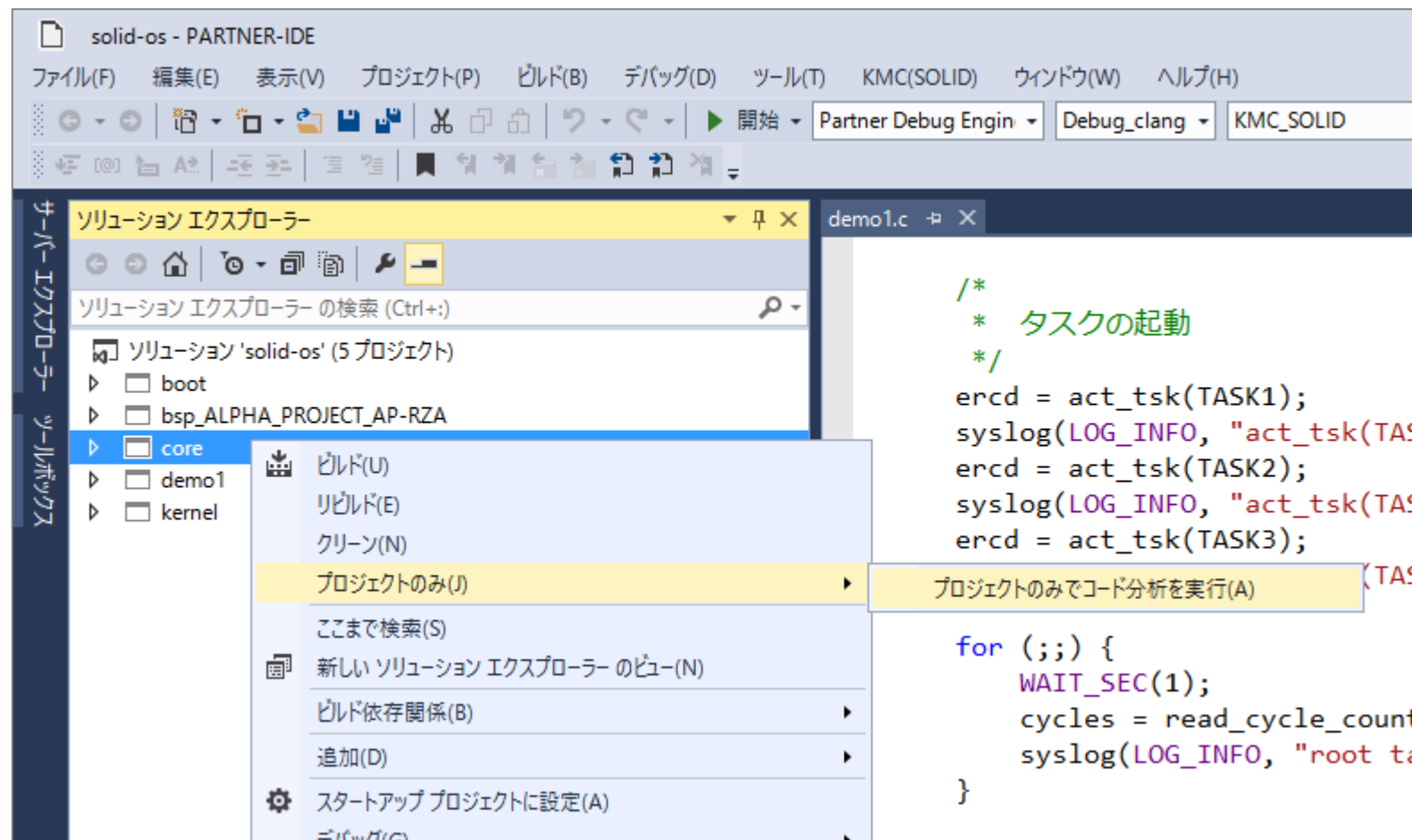
SOLID

- ソースコード**静的解析機能**を標準搭載
 - 追加設定なく、ビルドの設定さえあれば、解析可能
 - 0除算や不正ポインタアクセスの可能性などを検出

バグの自動検出も 可能な新開発環境

SOLID

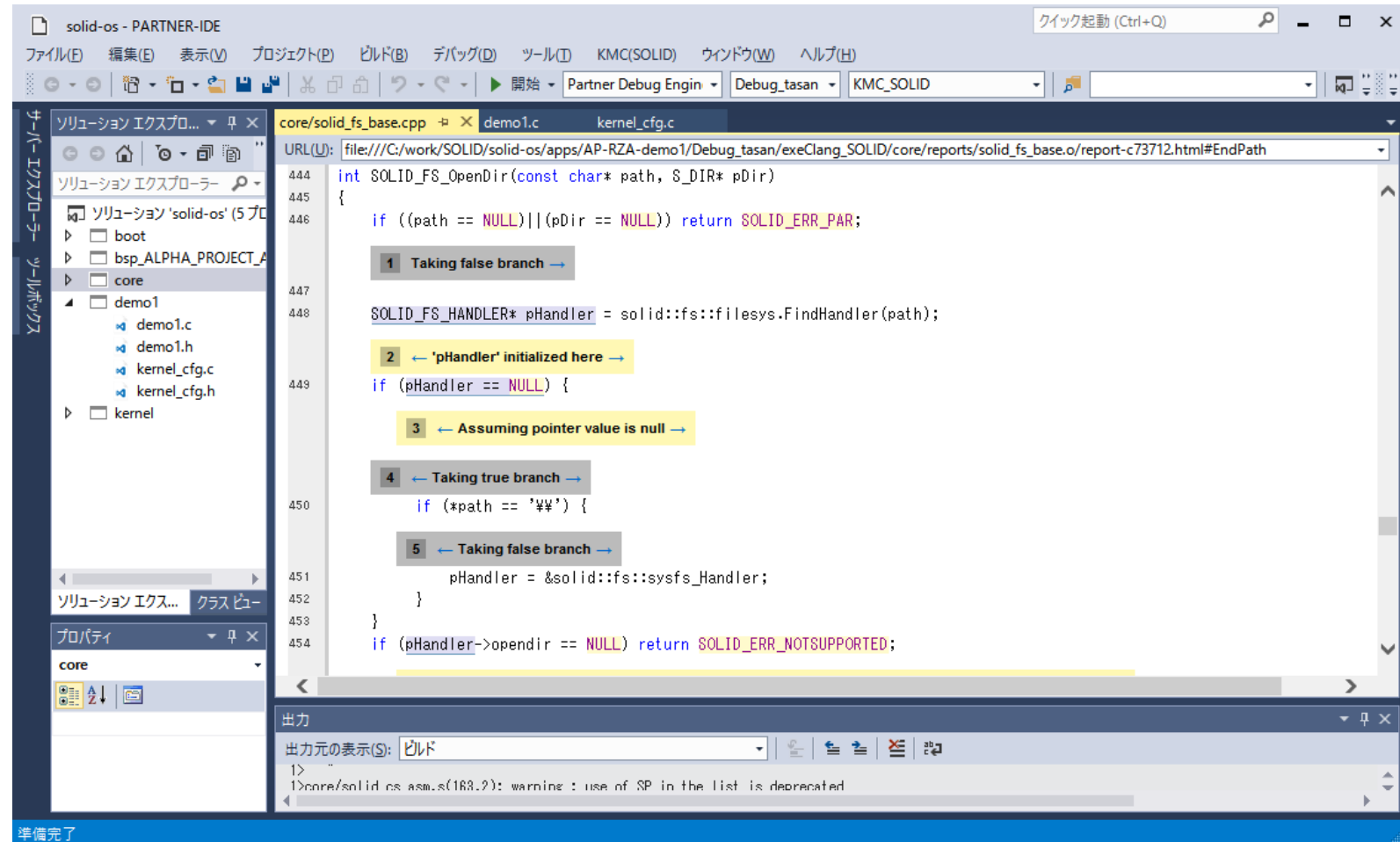
ソース静的解析も、
メニューからワン
クリックで実施



バグの自動検出も 可能な新開発環境

SOLID

実行前のバグ検出
が可能。
デバッグを効率良
くします



バグの自動検出も 可能な新開発環境

SOLID

- MMUとリンクスクリプトが連携する**メモリプロテクション**
 - スタックの突き抜けもプロテクション
 - プログラム的に存在しない領域や、データ領域の実行など、不正アクセスを即時に検出
 - 単一アドレス空間によるプロテクション機構、既存のRTOS開発モデルと同じ
- **アドレスサニタイザ**による実行時メモリバグ検出機能
 - iOSアプリ開発環境などで定評の**アドレスサニタイザ**を、組み込みで初めて実現
 - バッファオーバーランや、解放後のメモリアクセスなど、**不正アクセスを自動検出**

バグの自動検出も 可能な新開発環境

SOLID

メモリアクセスバグの自動検出！！

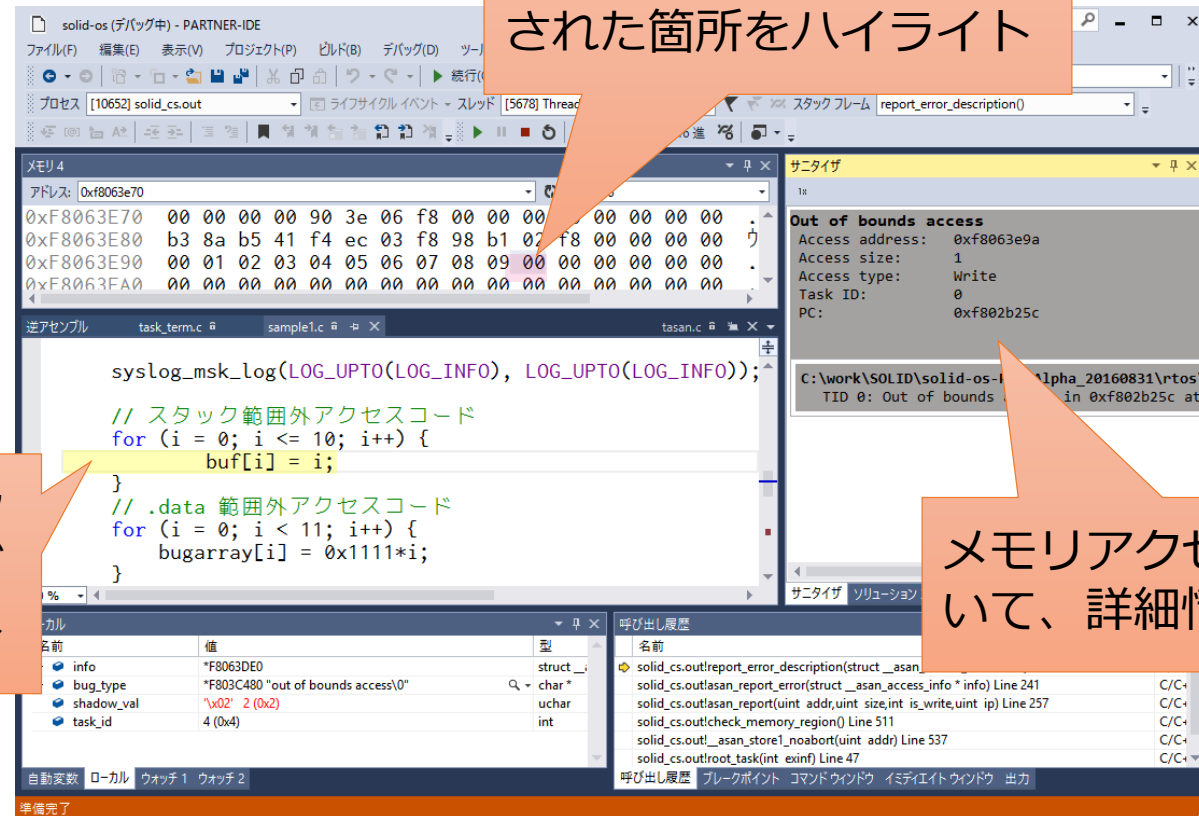
- ・バッファオーバーラン
- ・不正ポインタ
- ・二重解放など

手順はとてもシンプル
サニタイザモードでビルド&実行
するだけです

間違ったメモリアク
セスを行ったプログ
ラム行をハイライト

間違ったメモリアクセス
された箇所をハイライト

メモリアクセス違反につ
いて、詳細情報を表示



チーム開発の悩みを解決!!



分散開発に適した、アドレス解決可能なローダーを標準搭載。

バグの自動検出も 可能な新開発環境

SOLID

- **アドレス解決可能なELFローダー**を標準搭載

- モジュール間で外部シンボルを実行時に解決
 - リンカに機能追加と、専用ローダーを用意
 - ロードした追加ELFモジュールをアンロードする事も可能
- MMUと連携して、効率良くメモリを利用

- IDEとの連携

- メモリマップエディタとの連携
- デバッガとの連携

- メインモジュールはROMのまま、**追加ELFのみを改変**してソースデバッグが可能

分散開発に便利

バグの自動検出も 可能な新開発環境

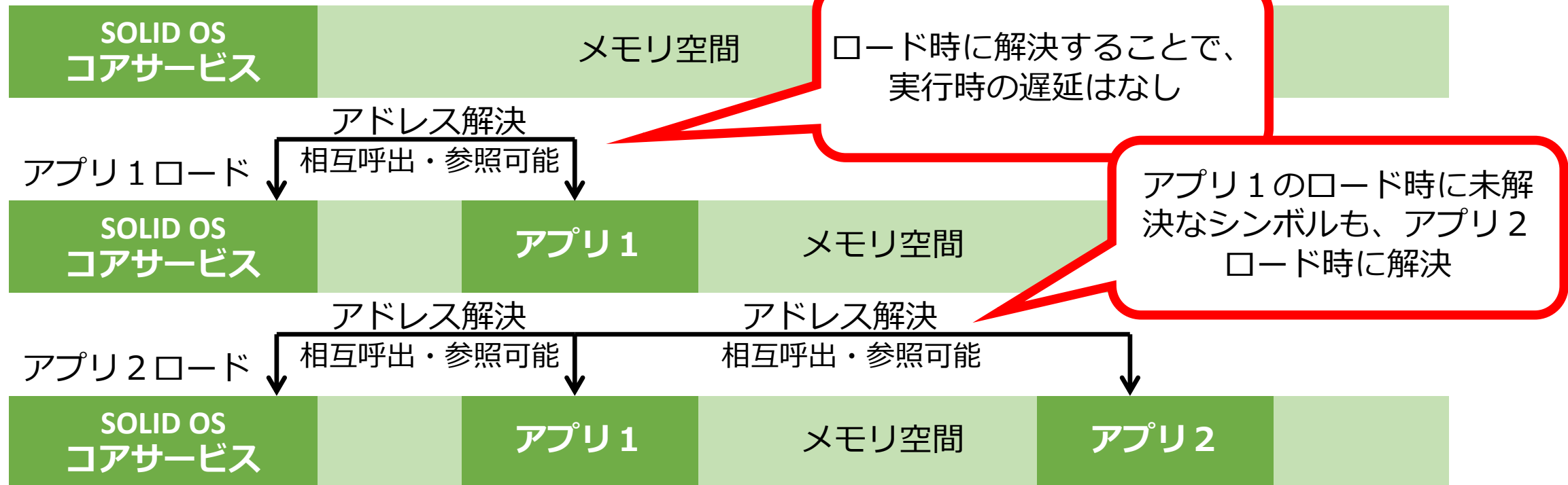
SOLID

組込み向き!!

単一アドレス空間で

静的リンクと同じような結果を実現するローダー

起動直後



バグの自動検出も 可能な新開発環境

SOLID

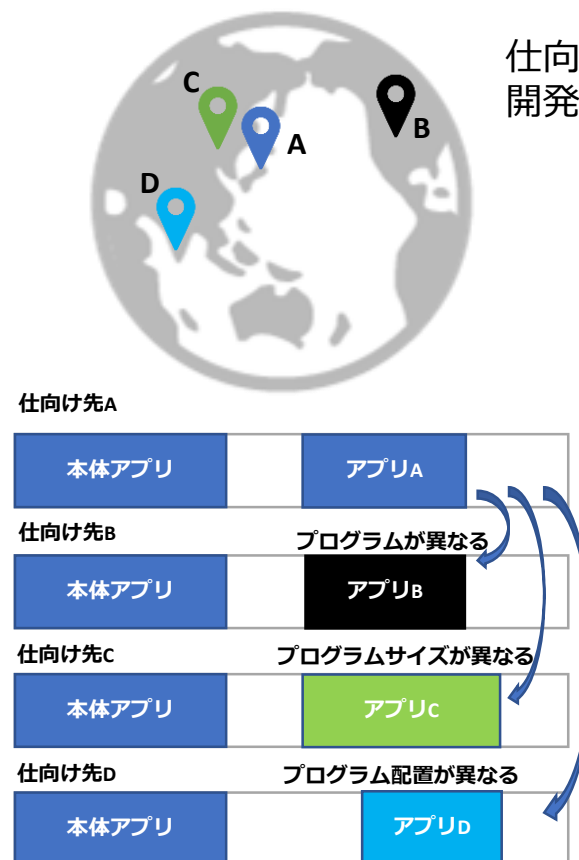
・分散開発への応用

開発協力会社がアプリケーション単位で
プログラムを分担して開発する場合



・仕向地向けソフトへの応用

仕向け先別にアプリを分けて
開発、格納する場合



まとめると



バグの自動検出も
可能な新開発環境

SOLID

**SOLIDは、
ITRON以上Linux未満を実現する開発環境**

従来のプログラムモデルのまま、
Linuxライクな効率良い開発が可能

RTOS/SOLIDで開発することのメリット

- より**省メモリ**で動作するので、RAMが少ないシステムで構築可能
 - 例えば、RZ/A1Hの10MByte内蔵RAMだけでも、RTOS本体にTCP/IPやFATファイルシステムなどをつけても、余裕で動作可能
 - 大容量のDDRを使わなくても、**内蔵RAMやSDRAMで動作**させることが可能
すなわち、基板作成コストや部品選定コストが小さくなる
- 実時間処理を、より低速のプロセッサで実現可能
 - **省電力性や応答性にすぐれるシステム**開発が可能
 - 標準カーネルのASP3はティックレスなので、割り込み負荷が小さい
 - SOLID-OSは多重割り込み対応

RTOS/SOLIDで開発することのメリット

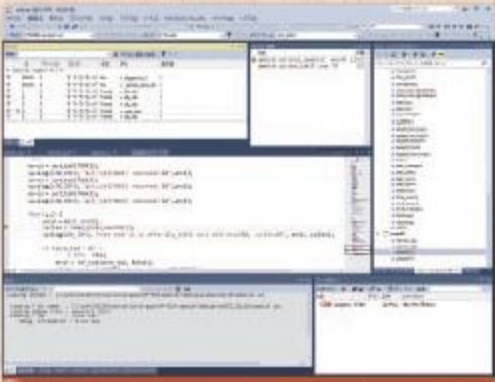
- ローダーを使って**機能単位の作成・実行が可能**
 - ファームウェアのアップデート機能の作成が容易
 - ネットワーク越しにアップデートする機能などの実装も簡単
 - 仕向け地向けに異なる実行モジュールを用意することが簡単
 - 起動時に仕向け地のモジュールを読み込むだけ！
- より**堅牢なソフトウェア**開発が可能
 - サニタイザによるメモリバグの洗い出し
 - メモリ破壊した時点で停止でき、素早く問題解決が可能
 - メモリプロテクションによる、バグ検出
 - SOLID-IDEとの連携で、実行メモリは書込み不可など、適切なプロテクションを自動設定
 - スタックもサイズオーバーはプロテクション機能で検出
 - より安全なソフト開発が可能に！

SOLIDとルネサスCPU

- 多くのRZファミリに対応します
 - RZ/A1プロセッサを、SOLID標準開発プロセッサとして利用しました
- RZ/Gシリーズ含め、Cortex-A搭載のRZファミリの動作確認実績があります
 - Cortex-A7搭載のRZ/Nに対応する事も可能
- Cortex-Aプロセッサであれば利用可能
 - 64bit Arm v8-Aプロセッサの32bitモードにも対応済
 - SMP構成のCortex-Aプロセッサにも対応済
- Cortex-Aでなく、Cortex-R/M環境でも利用可能です
 - MMU対応機能は動作しません
 - サニタイザやローダについては、利用可能な場合もありますが、制限があります

¥50,000-で、SOLIDを体験してみよう

SOLID スターターキット



SOLID-IDE, コンパイラ
RTOS 込みの付属ボード用BSP



PARTNER-Jet2 Model10



Renesas RZ/A1H 評価ボード

IDE,コンパイラ,JTAG,ボードと、PC以外の必要な物全てが含まれます。
イーサネットドライバなどのBSPも付属して、購入してすぐにSOLIDの開発環境を体験できます!!

[スターターキットWEB →](#)

ルネサス様でのSOLIDセミナー開催

「ITRON以上、Linux未満！ 組み込みOS開発の新提案、 京都マイコンSOLID体験セミナー RZ/A1H編」

ご好評につき、昨年に引き2月も開催いたします。

- 実開発環境を触って体験する、ハンズオン形式でのセミナー
- 先着申し込み順、受講費用は無料
- 2018/2/14 東京
- 2018/2/21 大阪

ルネサス様HPから
お申込みください



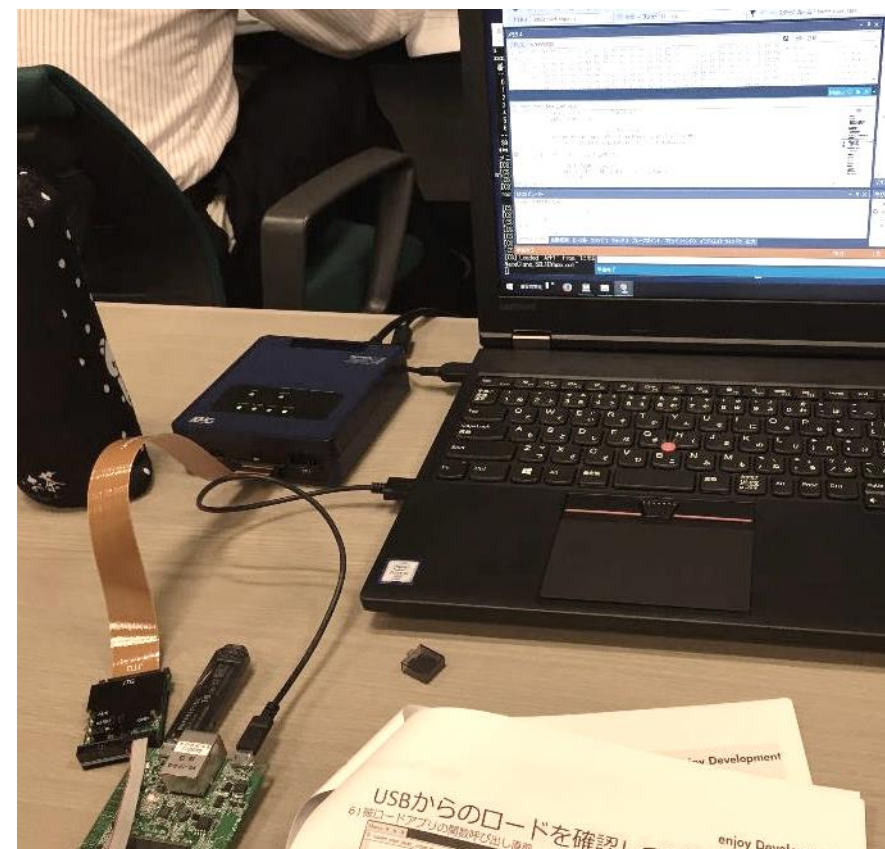
SOLID体験セミナー

検索

<https://www.renesas.com/ja-jp/support/training/seminar/individual/products-seminar/20170830.html>

セミナーの様子

セミナー実習用のPC、ボード、エミュレータを1式／1名使って、じっくり操作体験をしていただきます。

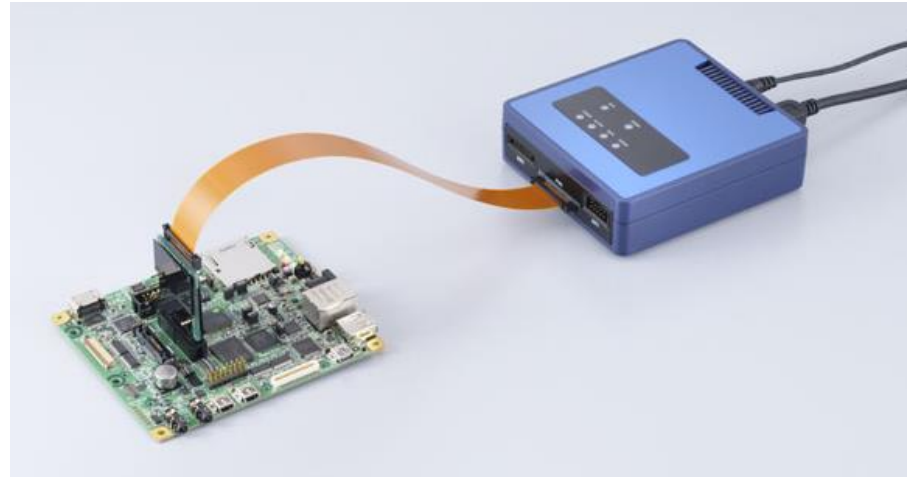


PARTNER-Jet2

Arm v8プロセッサ 64bit対応デバッグについて

PARTNERJet2

4Gバイト/8Gバイトトレースメモリ
搭載の高機能モデル
PARTNER-Jet2 Model 20/30



PARTNER-Jet2 Model 10
USBバスパワー対応で、
低価格で導入しやすい普及モデル

PARTNER-Jet2の利便性

- USBバスパワー対応

- PARTNER-Jet2 Model10は、USBバスパワーで動作
- USB3.0のみならず、多くのPCではUSB2.0でも動作可能
 - 電圧低下時はLEDでアラーム表示（その場合はACアダプタをご利用下さい）



- プローブホットプラグ対応

- ターゲット動作中に、後からプローブを接続してデバッグを開始したり、取り外したりする事が可能
- 障害が発生してからJTAGを接続し、システム状態の確認が可能
- ArmプロセッサとIntelプロセッサで対応



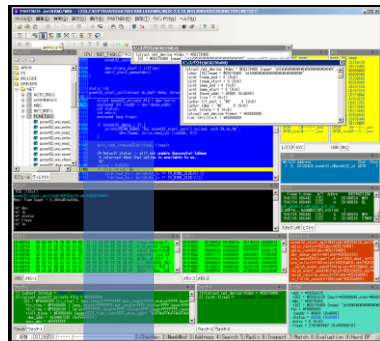
PARTNER-Jet2

Arm v8 64bit対応 (デバッグ機能)

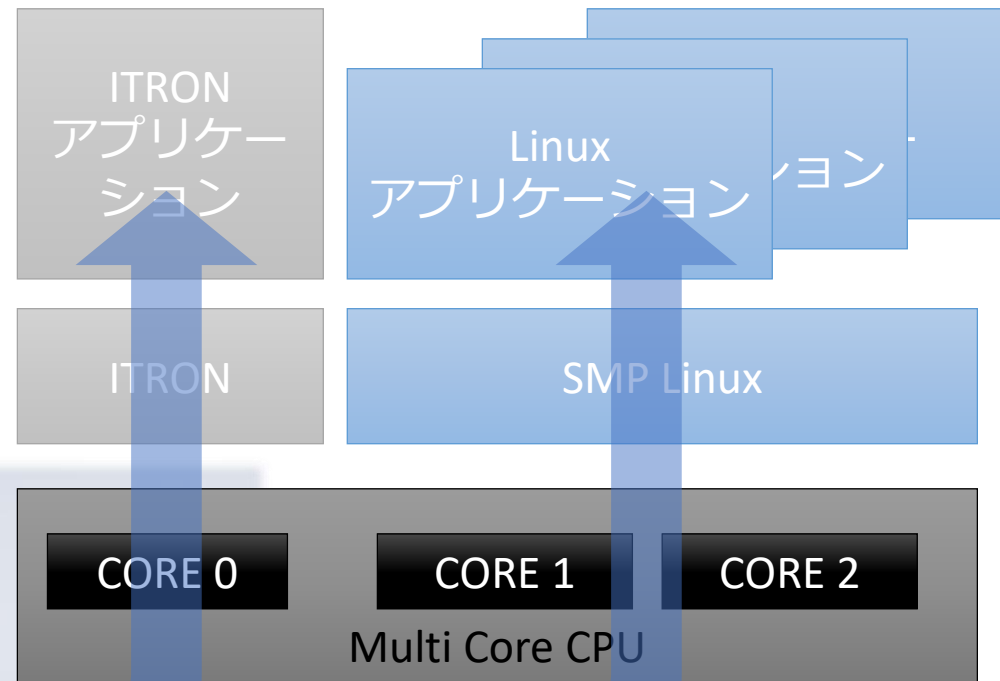
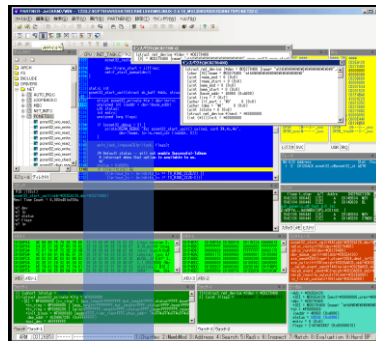
- 従来の32bitのArmプロセッサに加え、64bitプロセッサにも対応しています
- 対応済み64bitプロセッサ
 - Cortex-A53, Cortex-A57, Cortex-A72, A73, A35
- 64bit/32bit混在のデバッグも可能
 - ブレーク時に実行命令のモードを自動認識します
- v8Aプロセッサの状態にも対応
 - EL0,EL1,EL2,EL3のExceptionモード
 - Secure/Non-Secureモード
 - 仮想マシンモード
 - ハードウェアブレークなどを、モードを条件追加して設定可能
 - 実行しているモードなどをわかりやすく表示
- LinuxやLinux+RTOSなど、マルチOSシステムのデバッグができます

ITRONとSMP Linuxを同時にデバッグ

ITRONのデバッグ



Linuxのデバッグ



一台のPARTNER-Jet2で二つのOSを同時にデバッグ

ありがとうございました

Kyoto Microcomputer Co.,Ltd.

<http://www.kmckk.co.jp>

展示コーナーへも是非お立ち寄りください