

1

ソフトウェア開発プラットフォーム

SOLID

京都マイクロコンピュータ 2017年 提供予定新製品

SOLID

かしこく開発・スマートにデバッグ

プログラムの問題を自動検出, RTOS とコンパイラを一体化した
革新的開発プラットフォーム

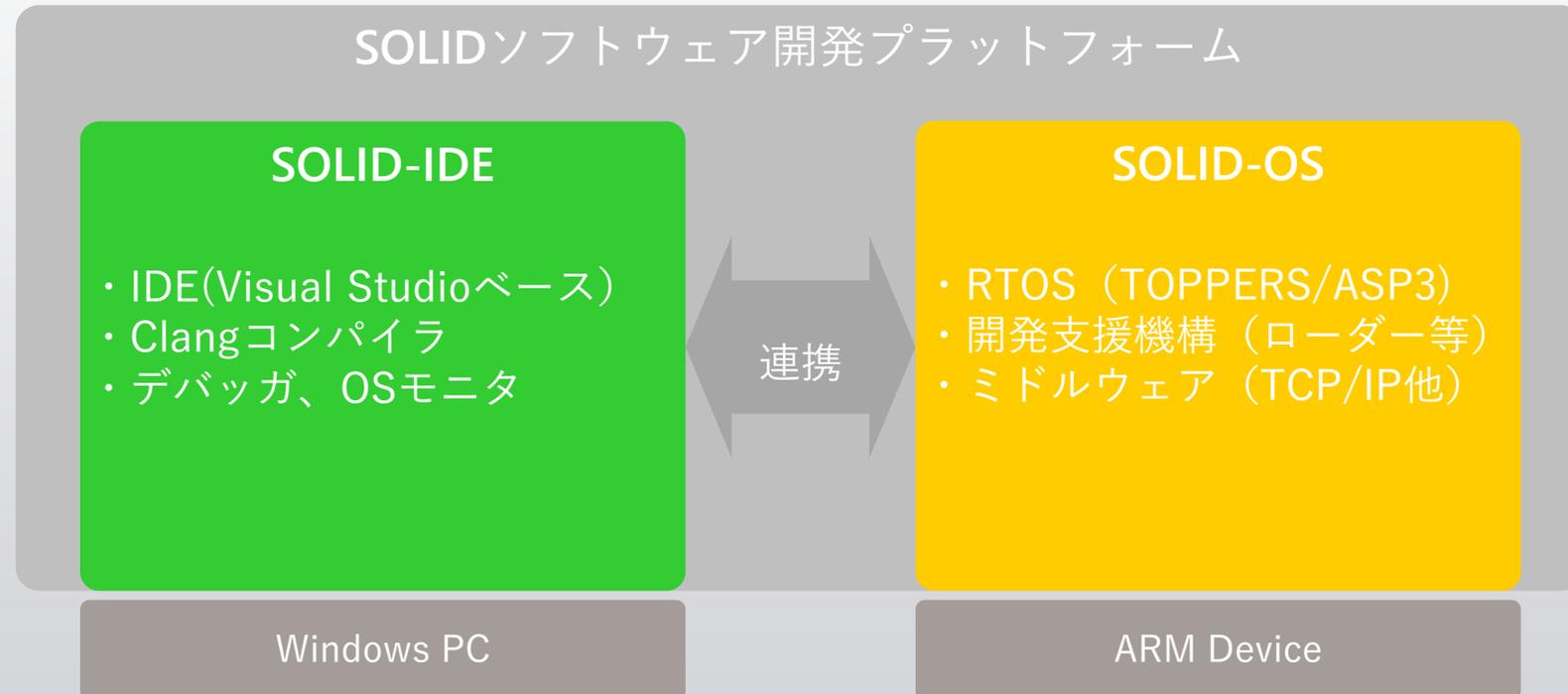
SOLIDが目指すもの

- ▶ 「ツール屋」であるKMCが、デバッガやコンパイラを提供してきた経験をもとに、開発者に”楽しく”, ”快適に”開発に取り組めるソフトウェア開発プラットフォームを提供する
- ▶ 開発者が本来のクリエイティブな業務に最大のパフォーマンスを発揮して「かしく開発」し、また、不具合のモグラ叩きに翻弄されずに「スマートにデバッグ」する環境を提供する
- ▶ 組み込み機器の開発手順をシンプルに変える

SOLIDの構成

- ▶ SOLIDは組み込み用「リアルタイムOS」と「開発ツール」を一体化したソフトウェア開発プラットフォーム
 - ▶ リアルタイムOSには、TCP/IPなども付属する
 - ▶ 開発ツールには、IDEやコンパイラ・デバッガが含まれる
- ▶ SOLIDは、RTOS含んだSOLID-OSと、統合開発環境SOLID-IDEから構成される

SOLIDの構成



SOLIDの構成

SOLID-OS

- ▶ SOLID-OSとして、名古屋大学を中心としたTOPPERSプロジェクトで開発されたオープンソースカーネルである**TOPPERS/ASP3**を採用します。
 - ▶ 組み込み機器として実績のあるμITRON 4.0仕様準拠
 - ▶ ASP3のティックレス仕様により実行効率・電力効率が良い割込み制御が可能
 - ▶ SOLID-OSとして
 - ▶ カーネル本体、プロセッサ依存部、BSPを提供
 - ▶ TCP/IP, Fileシステム、スクリプトエンジン等のミドルウェアも提供
 - ▶ ロイヤリティフリー

かしくく開発

SOLID-IDE

- ▶ 開発準備から性能評価までの作業をシンプルに変えるため、SOLID-IDEはユーザーインターフェースに定評のある**Visual Studio**をベースに独自開発しました。
 - ▶ コンパイルからデバッグまで、全てWindows上で作業できるので、「ビルド→転送→実行→デバッグ」、の手順がシンプル
 - ▶ Intellisense 機能（エディタでのコード補完）をはじめ、Visual Studio の多くの特徴的な機能が使える
 - ▶ Visual Studio Shellをベースに独自に開発したIDEは、ロイヤリティフリー
 - ▶ Clangコンパイラで検出したエラーを、IDE上で分かりやすく表示
 - ▶ 静的解析、動的解析ともに、問題箇所をIDE上に表示

かしかく開発

Clangコンパイラ

- ▶ SOLIDプラットフォームでは、静的解析ツール、動的解析ツール機能を豊富に備えた**LLVM/Clangコンパイラ**を採用します。
 - ▶ LLVM/Clangは次世代のコンパイラとして、利用されはじめている
 - ▶ MacOS/iOS開発環境の標準コンパイラ、ARM Compiler 6、FreeBSDの標準コンパイラなど、多くの分野で標準的に利用されている
 - ▶ Clangコンパイラは、オプションや言語仕様拡張など、GCCコンパイラとの互換性が高い
 - ▶ ビルド時に静的解析ツールとしてClangを使用することにより、「未初期化変数の利用」「メモリリーク（解放もれ）パスの検出」などが検出可能
 - ▶ 実行時に「アドレスサニタイザ」などデバッガと連動した動的解析が可能

かしこく開発

Clangコンパイラ 静的解析

```
17 int main()
18 {
19     vector<int> vect;
20     char *str = (char *)malloc(100);
21     if (str == NULL) {
22         return -1;
23     }
24     vect.push_back(4);
25     vect.push_back(1);
26     vect.push_back(10);
27     vect.push_back(-3);
28     vect.push_back(1);
29     vect.push_back(-5);
30
31     print(vect);
32     sort(vect.begin(), vect.end());
33     print(vect);
34     sort(vect.begin(), vect.end(), greater<int>());
35     print(vect);
36
37     return 0;
38 }
```

1 Memory is allocated →

2 ← Taking false branch →

3 ← Potential leak of memory pointed to by 'str'

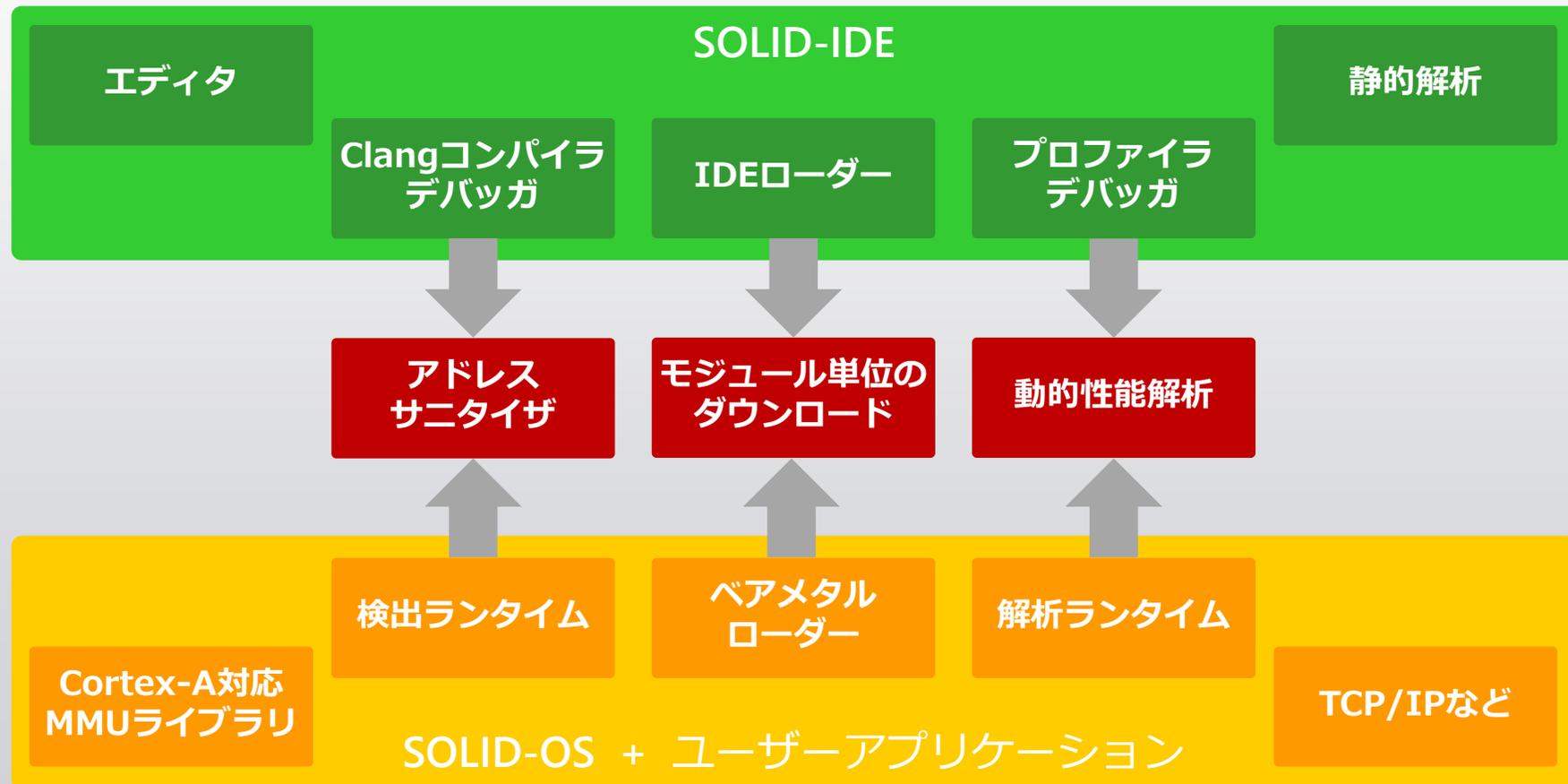
かしこく開発

シンプルベアメタルローダー

- ▶ RTOSやベアメタル環境では、複数拠点などでの分割開発時に、個別に分割したプログラム単位でのローディングや実行が煩雑になっている
- ▶ 新規にMMU対応のシンプルベアメタルローダーを開発
 - ▶ MMUは仮想アドレスを使うが、シンプルに使えるよう単一空間のみ
- ▶ 複数に分割されたプログラムでも、それぞれ個別に開発・ローディング可能
 - ▶ 分割ローディング時のアドレス解決機構を搭載
 - ▶ メモリ利用効率が低下しないよう、MMUを使ってアドレス割り付け
 - ▶ MMUのプロテクションを有効にし、デバッグ効率も向上
 - ▶ IDEと連携したローディング
 - ▶ アップデートなどの効率化

スマートにデバッグ

SOLID-IDEとSOLID-OSの密結合



スマートにデバッグ アドレスサニタイザ

組み込み初登場！アドレスサニタイザ

▶ アドレスサニタイザとは？

- ▶ LLVM/Clang コンパイラのテスト支援機能で、メモリ破壊やリークなどを実行時に検出
- ▶ iOSアプリケーション開発環境のXcodeで使えることで有名

<https://developer.apple.com/xcode/jp/>

▶ 簡単に使える

- ▶ SOLID-IDEでビルド・実行モードを「アドレスサニタイザモード」に設定するだけ
- ▶ 事前にバグがありそうな箇所などの検討必要なし
- ▶ 実行するだけで、間違ったメモリアクセスを自動的に検出します

SOLID-OSとSOLID-IDEが連携するので、わかりやすく、簡単に使えます

スマートにデバッグ
アドレスサニタイザ

デモンストレーション

スマートにデバッグ アドレスサニタイザ

間違ったメモリアクセス
された箇所をハイライト

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

メモリアクセス違反につ
いて、詳細情報を表示

メモリ4

アドレス	0xf8063e70	0xf8063e80	0xf8063e90	0xf8063fa0
0x	00 00 00 00 90 3e 06 f8 00 00 00 00 00 00 00 00	b3 8a b5 41 f4 ec 03 f8 98 b1 02 f8 00 00 00 00	00 01 02 03 04 05 06 07 08 09 00 00 00 00 00 00	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

```
syslog_msk_log(LOG_UPTO(LOG_INFO), LOG_UPTO(LOG_INFO));  
  
// スタック範囲外アクセスコード  
for (i = 0; i <= 10; i++) {  
    buf[i] = i;  
}  
  
// .data 範囲外アクセスコード  
for (i = 0; i < 11; i++) {  
    bugarrray[i] = 0x1111*i;  
}
```

ローカル

名前	値	型
info	*F8063DE0	struct _...
bug_type	*F803C480 "out of bounds access(0"	char *
shadow_val	"\x02" 2 (0x2)	uchar
task_id	4 (0x4)	int

呼び出し履歴

名前	詳細情報
solid_cs.out!report_error_description(struct _asan_acc...	
solid_cs.out!asan_report_error(struct _asan_access_in...	
solid_cs.out!asan_report(uint addr,uint size,int is_write,uint ip) Line 257	C/C+
solid_cs.out!check_memory_region() Line 511	C/C+
solid_cs.out!_asan_store1_noabort(uint addr) Line 537	C/C+
solid_cs.out!root_task(int exinf) Line 47	C/C+

準備完了

2017年リリース 対応プロセッサ

- ▶ ARM® Cortex®-A9 プロセッサに代表される、ARMv7アーキテクチャのマイクロプロセッサを搭載したシステムを1st ターゲットとします。
 - ▶ 組み込み機器向けの汎用プロセッサおよびSoCにおける採用例が多い
 - ▶ 性能 vs 消費電力 に最適化されたコア
 - ▶ MMUによるメモリプロテクションや仮想アドレスの使用が可能
 - ▶ 標準となるハードウェア（評価ボード）用のBSPを提供
 - ▶ ハードウェアがなくても、シミュレータですぐに動作確認が可能

*今後 ARM Cortex-Mx プロセッサも対応計画中

2017年リリース 対象分野

- ▶ SOLIDプラットフォームの開発ターゲット分野は、リアルタイム制御を要求される組み込み機器です。
 - ▶ メカ制御、センサ入力処理や信号処理といった、リアルタイム性が必須の組み込み機器用途
 - ▶ ARM Cortex-A9クラスのプロセッサ性能を要求される高速処理システムであって、Linuxを実装するためのメモリ資源確保が難しい場合
- ▶ 対象分野
 - ▶ 産業・家電機器
 - ▶ アミューズメント機器
 - ▶ デジタルイメージング機器など

2017年リリース

2017年度 1Q提供予定

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

enjoy **D**evelopment

ありがとうございました

Kyoto Microcomputer Co.,Ltd.

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

KMG Inc.

<http://www.kmg-inc.jp>