

組込システムにおける C++の利用

2010-10-23 @kikairoya

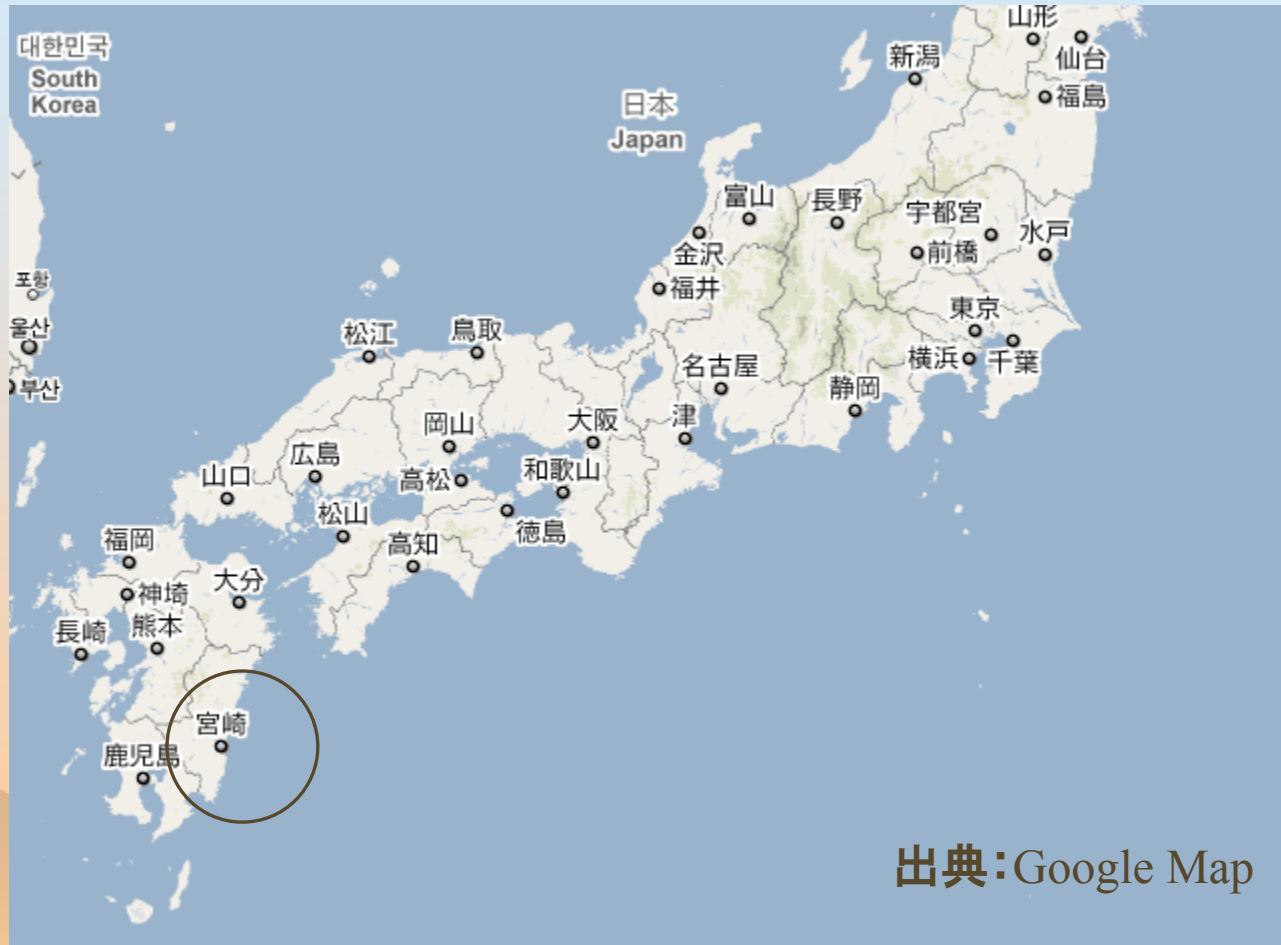


発表者について

- ❁ **名前:** @kikairoya / id:kikairoya
- ❁ **業種:** 精密機械設計・電子回路設計
- ❁ **勤務地:** 宮崎県都城市
- ❁ **実年齢:** 満22歳



宮崎県こんなところですよ



出典: Google Map

宮崎県こんなところですよ

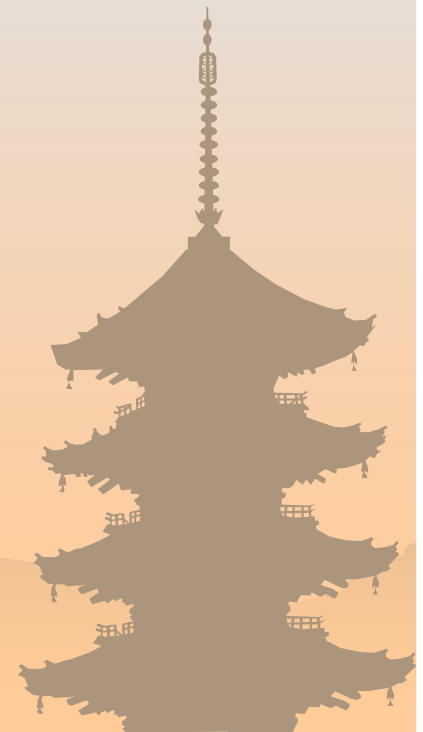
- ❁ 宮崎-大阪間
直線距離：約500km
陸路：約900km
- ❁ 人間より鶏が多い
- ❁ 住宅街で養豚
- ❁ 複線の鉄道が無い
- ❁ 東九州道早く作れ



今日話すこと

- ❁ 「組込」ってどんな環境？
- ❁ 組込でもC++使えるの？
- ❁ 組込用にチューニングされたライブラリ
- ❁ ntfmtの紹介
- ❁ まとめ

※Boost成分はほとんどありません



組込ってどんな環境？

- ❁ 定義は人によってマチマチ
- ❁ 乱暴に表現すると、
「セルフ開発が出来ない環境」
- ❁ 「組込はリソース制限が厳しい」は
必ずしも真ではない
 - 一昔前のPCより今のTVレコーダのほうが
性能がいい



組込システムの例

🌀 リモコン



組込システムの例

- ❁ リモコン
- ❁ 炊飯器



組込システムの例

- ❁ リモコン
- ❁ 炊飯器
- ❁ 自動販売機



組込システムの例

- ❁ リモコン
- ❁ 炊飯器
- ❁ 自動販売機
- ❁ 自動車



組込システムの例

- ❁ どれも機器上でコンパイラを動かさない
 - Linux載せれば動かせるだろうけれど普通しない
- ❁ 基本的にエンドユーザは自分でプログラムを追加しない
 - 携帯電話・家庭用ゲーム機は例外
- ❁ 使用できるリソースはモノによってピンキリ
 - クロック: 数kHz - 数GHz
 - RAM: 数バイト - 数ギガバイト



組込でもC++使えるの？

- ❁ GCCとnewlibが移植されていれば、C++のほぼフルセットが使えます。
- ❁ TemplateベースのSTLコンテナも使えます。
- ❁ ROMさえ足りれば、例外もRTTIも使えます。
- ❁ スレッド周りは結構怪しいです。



組込でもC++使えるの？

- ❁ 一部のベンダはEDGをベースにしたC++コンパイラを提供しています。
- ❁ 一部のベンダはEC++をベースにしたC++ライブラリを提供しています。
- ❁ PICはC++コンパイラがありません。AVRやR8Cなどを使いましょう。



組込でもC++使えるの？

- ❁ C++だからと言って、例外やらRTTIやらを使う必要はありません。
- ❁ Templateやデストラクタはノーコストで使えるので、大いに活用しましょう。
- ❁ C++0xの言語機能は普通に使えます。
 - auto/Lambda/Variadic Templates/
rvalue-reference/deleted functions



組込でもC++使えるの？

- ❁ 多くのBoostライブラリは問題なく使えます。
 - Array/Foreach/Optional/Operators/Unitsなど。
- ❁ 一部のBoostライブラリは例外が必要です。
 - Smart-Pointers/Function/Variantなど。
- ❁ RTTIが必要なBoostライブラリもあります。
 - Any/Asioなど。
- ❁ Functionが例外を要求するとかwtf



組込向けにチューニングされた ライブラリ

❁ 「組込向けの要件」

– 省メモリ(特にRAM)

– Genericな実装

- ターゲットごとにintの表現が違う
サイズ、バイトオーダー、ビットオーダー、アライン

– 例外フリーな実装

- 基本的にマルチスレッド(のようなもの)で動くため、
飛んでも取れないことがある
- そもそも例外機構がROMに収まらないことも



組込向けにチューニングされた ライブラリ

❁ newlib

- コンパクト(自称)なlibc
- C99フル実装なので必要以上にでかい
- たまにsizeof(int)==4を仮定している場所がある
- 再入安全には非常に気を使っている

❁ libstdc++

- GNUのC++ライブラリ
- 例外を使わなければゼロオーバーヘッド



ntfmt (New Type-safe formatter)

- ❁ 組込環境ではsnprintfが(サイズの関係で)使えないことがある
 - newlibの場合、数十キロバイト必要
- ❁ iostreamはもっと使いにくい
 - 仮想継承やRTTI、例外機構で数百KiB必要
- ❁ →組込でも使いやすい、新しいformatterを作ってみた



ntfmt (New Type-safe formatter)

❁ 基本コンセプト

- 例外を飛ばさない
- 型安全・拡張可能
- 浮動小数点処理を切り離せる

❁ <http://github.com/kikairoya/ntfmt>

❁ 名称募集中です



ntfmt (New Type-safe formatter)


- ❁ 文字列中にフォーマット指定を埋め込まない
 - C++はコンパイル時に文字列を扱えないため
 - 指定と引数が違う時に例外を飛ばしたくない
- ❁ 動的メモリ確保をしない
 - RAMの使用量は予測可能にしておきたい
- ❁ テンプレートベースの実装
 - ユーザ定義型も出力できるように



ntfmt (New Type-safe formatter)

❁ 基本的な使い方

```
// standard form:  
sink_stdout() << -3839 << '¥n' ;  
sink_stdout() << fmt(853.842223, "%3.5e" ) << '¥n' ;  
// C++0x-mode form:  
sink_stdout().format(  
    fmt(-0x12f, "%+#10.5x" ),  
    '¥n'  
);
```



ntfmt (New Type-safe formatter)

✿ いろいろな出力先

```
char buf[100];  
sink_strbuf(&buf) << "to char-array" ;  
sink_strbuf(&buf[0], 100) << "to char *" ;
```

```
string s;  
sink_string(s) << "to std::string" ;  
stringstream ss;  
sink_stream(ss) << "to std::ostream" ;
```



ntfmt (New Type-safe formatter)

✿ ユーザ定義型

```
namespace ntfmt {
    template <class Fn, class T, class U>
    void ntfmt_printer(
        Fn &fn, pair<T, U> const &value, flags_t flags
    ) {
        fmt(value.first, flags).print(fn);
        fn( ',' );
        fmt(value.second, flags).print(fn);
    }
}

int main() { sink_stdout() << make_pair(10, 20); }
```



ntfmt (New Type-safe formatter)

❁ コードサイズ比較

- printf(3)自体のサイズを比較 (I/Oルーチンを除く)

```
int main() {
    char buf[10];
#ifdef NTFMT
    sink_strbuf(&buf) << 3247.554;
#else
    snprintf(buf, 10, "%f", 3247.554);
#endif
    puts(buf);
}
```



ntfmt (New Type-safe formatter)

⚙️ コードサイズ比較

- sh-elf-gcc-4.5.1 -m2 -Os -fno-exceptions
-fno-rtti -ffunction-sections -Wl,--gc-sections

NTFMT:

text:18600 data:2144 bss:112

STDIO:

text:29296 data:2208 bss:100

→ 10696バイト削減



ntfmt (New Type-safe formatter)

❁ コードサイズ比較

- printf(3)と依存ライブラリを比較 (I/Oルーチンを含む)

```
int main() {  
    char buf[10];  
#if NTFMT  
    sink_strbuf(&buf) << 3247.554;  
#else  
    snprintf(buf, 10, "%f", 3247.554);  
#endif  
}
```



ntfmt (New Type-safe formatter)

❁ コードサイズ比較

- sh-elf-gcc-4.5.1 -m2a -Os -fno-exceptions
-fno-rtti -ffunction-sections -Wl,--gc-sections

NTFMT:

text:13824 data:2144 bss:112

STDIO:

text:25520 data:2208 bss:100

→ 11696バイト削減



ntfmt (New Type-safe formatter)

- ❁ 例外・動的メモリ確保を使用せずに実装した
- ❁ そこそこ優秀な速度が出ている
- ❁ オブジェクトサイズは非常に小さい

- ❁ ユーザ定義型・ユーザ定義出力先の定義方法が美しくない
- ❁ カッコいい名前募集中



まとめ

- ❁ 組込環境≡クロスコンパイル縛り
- ❁ 「C++はCよりメモリ食う」は思い込み
- ❁ 標準ライブラリも普通に使えるよ
- ❁ 組込用にチューニングした
オレオレライブラリ作ってみたよ
- ❁ 問答無用で例外を要求する
`(std|boost)::function`マジwtf



おしまい

