

本スライドは、当日の講演資料の抜粋版です。

# よくわかる！定番！ 組込みソフトウェア開発技術の基礎

株式会社富士通ラーニングメディア  
西日本ソリューション部 松尾 圭浩

# Agenda

## 自己紹介・本日のトピック

### 1. 組込みソフトウェア開発を始めよう

- ・コンピュータを動作させるには、ソフトウェアが必要です。
- ・組込みソフトウェア開発では、クロス開発が一般的です。

### 2. メモリの使い方を確認しよう

- ・プログラムやデータの記憶場所としてレジスタとメモリが存在します。
- ・メモリのどこにデータが格納されているかはアドレスで管理
- ・データの格納方式の違いに注意する(エンディアン)
- ・メモリ構造
- ・実行中のデータを一時的に待避するスタック
- ・静的メモリと動的メモリ
- ・関数の引数は値渡し(Call by Value)
- ・スタックや動的メモリは無尽蔵に利用できない

### 3. 入出力デバイスを制御しよう

- ・入出力デバイスもアドレスで区別する
- ・2種類のI/O方式(CPUにより決まる)
- ・このスイッチの状態を読み取るには?
- ・ビットアサイン表を読めるようになろう
- ・ビットフィールドと共用体を用いて、ビット操作を拡張しよう
- ・ビットアサイン表に基づき定義してみると

### 4. 割込み処理で効率よく実行しよう

- ・ボタンが押されたら処理Aを実行したい場合、どうしますか?
- ・割込みにはいくつかの種類があります
- ・割込みを管理する割込みコントローラ
- ・割込みベクターテーブルを用いた割込み処理
- ・割込み要因ごとの実行処理を管理する割込みベクターテーブル

### 5. ROM化で仕上げ

- ・最終的にROMにプログラムやデータを配置する
- ・コンパイラがセクションに分け、リンクがメモリ空間に配置する

### 6. リアルタイムOSを使ってみよう

- ・OSを利用すると、実行制御が容易になる
- ・組込みでは、高いリアルタイム性能が実現できるリアルタイムOSが用いられる
- ・リアルタイムOS利用有無による比較
- ・プリエンプティブな優先度ベーススケジューリング
- ・基本的なタスクの状態遷移
- ・タスクスケジューリング
- ・システムコール

## まとめ

# 本日のトピック

組込みマイコンボードを用いて、電子オルゴールを作成しながら、組込みソフトウェア開発技術を基礎から学びます。

1. 組込みソフトウェア開発を始めよう！
2. メモリの使い方を確認しよう
3. 入出力デバイスを制御しよう
4. 割込み処理で効率よく実行しよう
5. ROM化で仕上げ
6. リアルタイムOSを使ってみよう

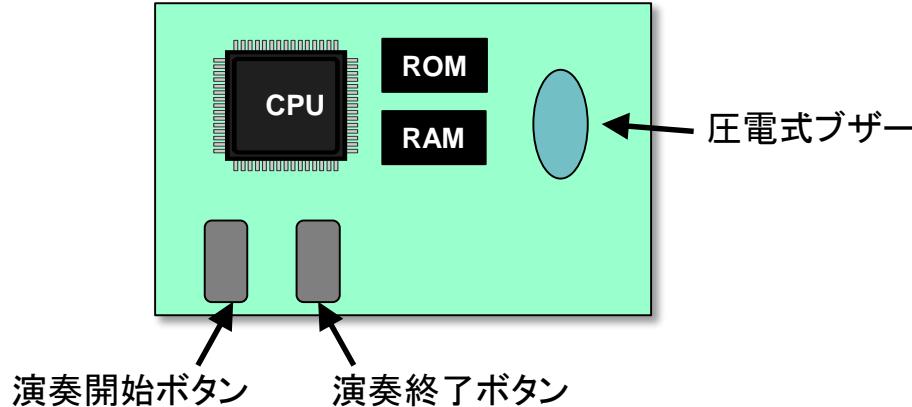
※本講演では、実装言語としてC言語を用いて説明します。

## 【参考文献】

- ・「標準テキスト組込みプログラミング～ハードウェア編～」
- ・「標準テキスト組込みプログラミング～ソフトウェア編～」  
(いずれも技術評論社出版、富士通ラーニングメディア共著)

# 1. 組込みソフトウェア開発を始めよう

- 以下のような組込みマイコンボードで音楽を奏でる電子オルゴールを作成してみよう！

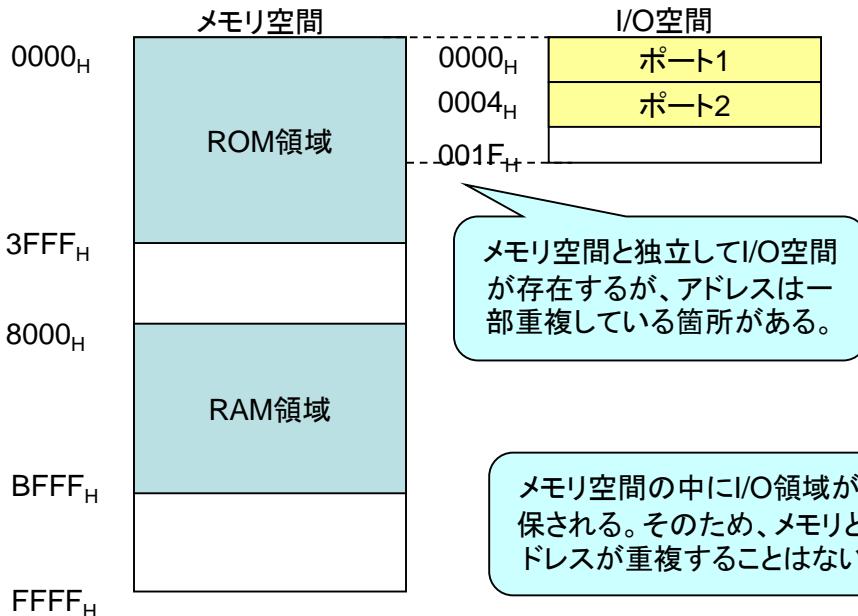


そもそも、この組込みマイコンボードにどのようにしてソフトウェアを作ればいいのでしょうか？

## ・2種類のI/O方式(CPUにより決まる)

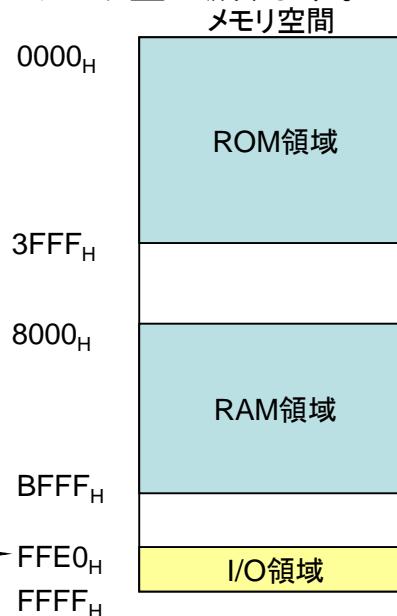
### 【I/OマップドI/O方式】

I/Oポートをメモリ空間とは別に用意した専用のアドレス空間で区別する方式です。



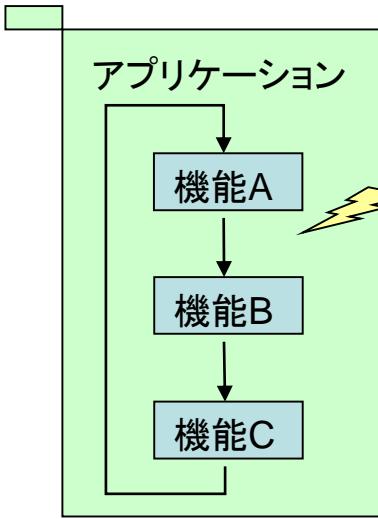
### 【メモリマップドI/O方式】

メモリ空間の一部にI/Oポートを区別する領域を設ける方式です。I/Oポート用に確保した領域分だけ、そのCPUで扱えるメモリのデータ量が減ります。

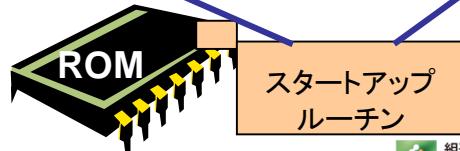


# ・OSを利用すると、実行制御が容易になる

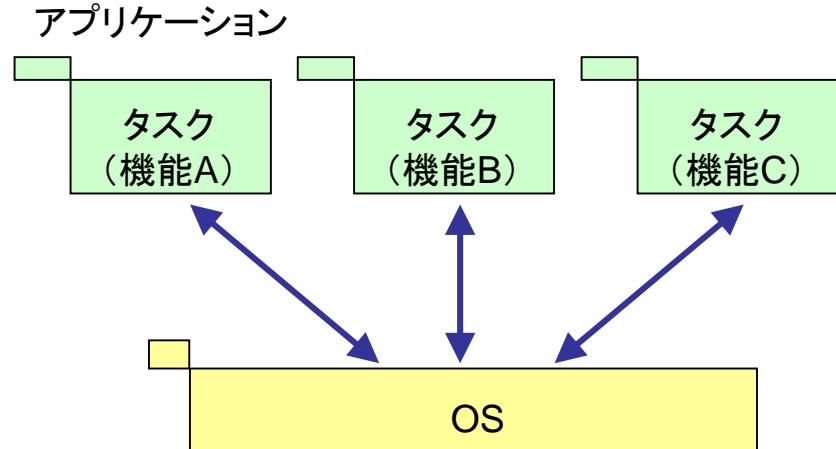
## 【OSが存在しない場合】



各機能の処理が長引くと他の機能の実行にも影響がでやすい



## 【OSが存在する場合】



複数のタスクを擬似的に同時並行に実行させることができ、実行タイミングの調整はOSが行う