

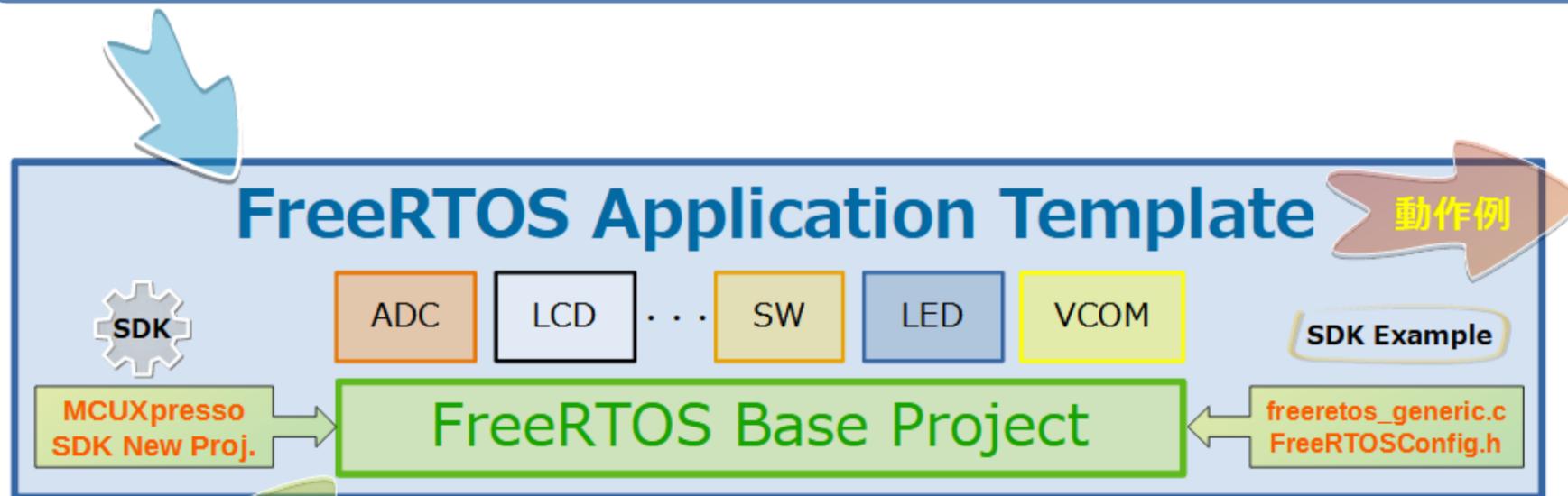
目的

- FreeRTOS基礎固め
- FreeRTOSアプリケーションプロトタイプ開発に適したプロジェクト  
ADC入力、LCD出力、SW入力（チャタリング対策済み）、LED出力、VCOM入出力 動作確認済み FreeRTOSプロジェクト

特徴

- 対象者：ベアメタル開発経験あり → FreeRTOS とベアメタル “差分説明” → ベアメタル視点でRTOS解説
- 豊富なベアメタルサンプルソフト活用が容易なテンプレートタスク構造 → FreeRTOSプロトタイプ開発に最適
- FreeRTOS とベアメタル 同一動作 2プロジェクト添付 → ベアメタルと RTOS 比較・評価 容易

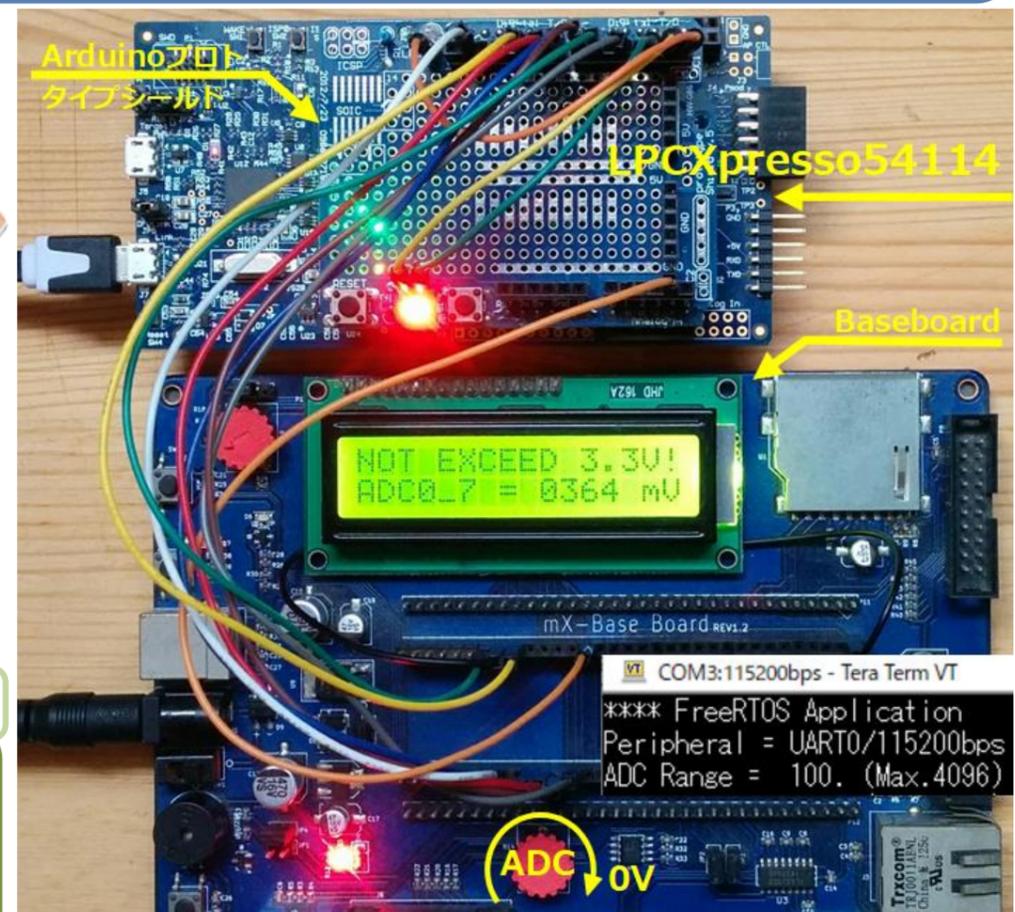
同一動作ベアメタルプロジェクトも添付



<https://www.freertos.org/Hardware-independent-RTOS-example.html>

FreeRTOS Base Project

- 原本：Hardware Independent FreeRTOS Example
- Semaphore と Queue 実装済み
- ハードウェア非依存（隠蔽）、開発タスク資産化、多人数開発 容易
- IoT MCUクラウド接続時、RTOS必須



FreeRTOSアプリケーションテンプレート + もくじ資料 → ¥2000（税込）販売中

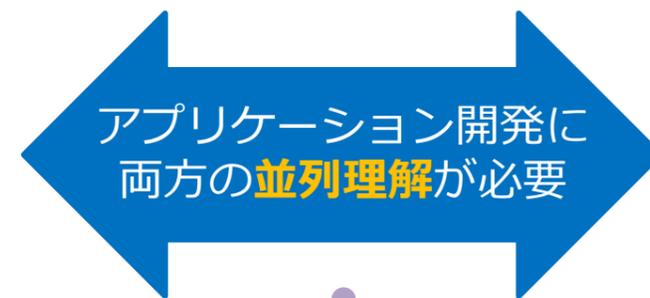
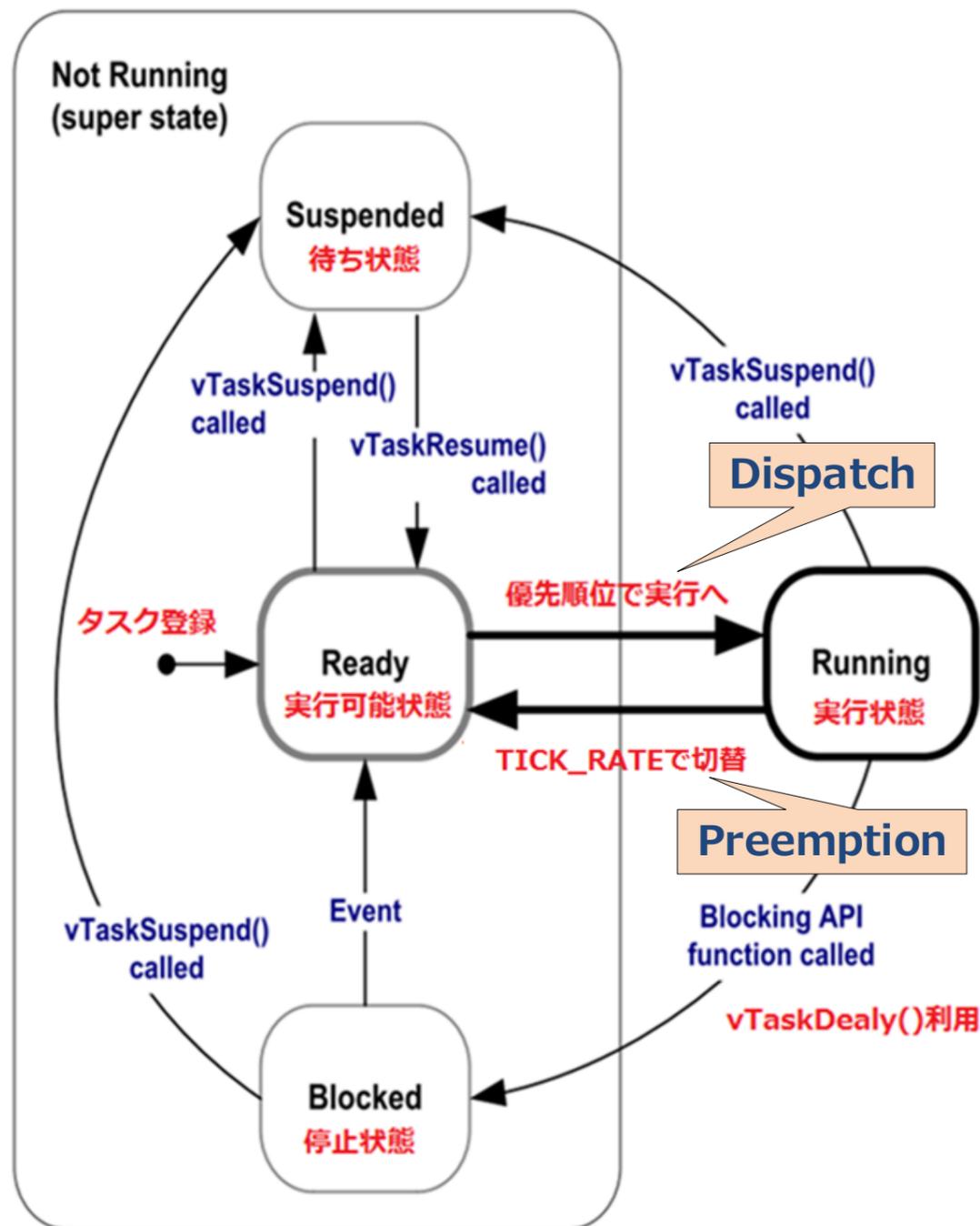
もくじ

テンプレート概要	1
FreeRTOS要点	2
アプリケーションテンプレ?	3
freertos_generic詳細	4
Base Project作成手順	5
BareMetal → RTOS	6
IoT MCU RTOS etc.	7
テンプレートタスク構成	8
低消費電力アプローチ	9
IoT Baseboard接続	10
参考資料と変更履歴	11

## FreeRTOS 要点 と ベアメタル視点の 習得アプローチ

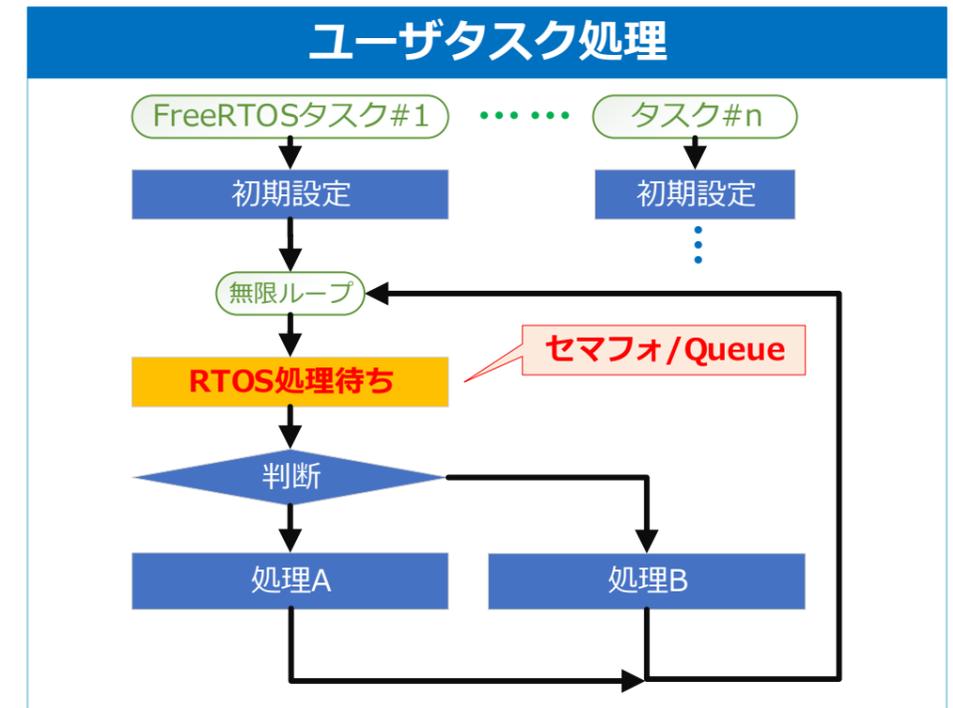
- RTOS (Kernel)が、複数タスクを**処理待ち**状態にし、処理待ちタスクの中から、**優先順位**に応じて1個のタスクを実行状態へ割当る
- Kernelが、優先判断、実行タスク切替え、タスク状態スタックプッシュ/ポップ処理を行い、その処理間隔が**TICK\_RATE**
- リアルタイムの**タスク切替え**には、高速なKernel動作、短周期のTICK\_RATE、タスク数に応じたスタック利用必須
- **FreeRTOSアプリケーションテンプレート**でRTOS状態遷移、RTOS処理待ち（セマフォ/Queue）、ユーザタスク処理のRTOS基本動作を理解し、より高度なミューテックス、イベントグループへステップアップするアプローチがお勧め。

### FreeRTOSのタスク状態遷移



ベアメタル/Serial から RTOS/Parallel へ 設計手法を変える必要

手間なく手法差を理解・実感できるツールが **FreeRTOSアプリケーションテンプレート**



テンプレート概要	1
FreeRTOS要点	2
アプリケーションテンプレ?	3
freertos_generic詳細	4
Base Project作成手順	5
BareMetal → RTOS	6
IoT MCU RTOS etc.	7
テンプレートタスク構成	8
低消費電力アプローチ	9
IoT Baseboard接続	10
参考資料と変更履歴	11

もくじ

1 : freertos\_generic

2 : FreeRTOS Base Project

3 : FreeRTOSアプリケーションテンプレート の関係

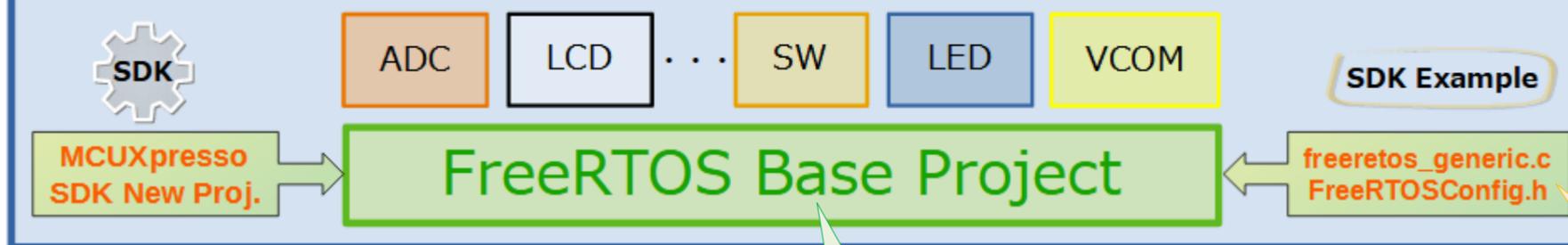
3

**FreeRTOSアプリケーションテンプレート** は FreeRTOS Base Project に

- ・プロトタイプ着手時必須な ADC、LCD、SW、LED、VCOM 各タスク を実装
- ・LPCXpresso54114 + Baseboard + Arduinoプロトタイプシールドで動作確認済み
- ・FreeRTOSプロジェクトと同じ動作の ベアメタルプロジェクト も添付

➔ **FreeRTOSプロトタイプ開発着手**に適し かつ ベアメタル開発経験者が **RTOS差分理解も容易** なテンプレート

## FreeRTOS Application Template



1

**freertos\_generic** はベンダ非依存の Hardware Independent FreeRTOS Example (原本) を

- ・NXP 向けにポーティング
- ・Semaphore と Queue の RTOS手段実装済み
- ・Hook関数 でRTOSヘユーザ独自処理追加が容易

※freertos\_generic詳細 P4参照

2

**FreeRTOS Base Project** は freertos\_generic の Hook関数 を利用し

- ・アイドルタスク時に 低電力動作 WFI 追加
- ・任意イベントセマフォ送付へ変更

※詳細は、Base Project作成手順 P5参照

### 注意事項

- ・本記載情報およびテンプレートソフトは、正確かつ慎重に作成したのですが、誤りがないことを保証するものではありません。
- ・万一、記載情報の誤り、またはテンプレートソフトに起因する損害がお客様に生じた場合においても、弊社は、一切その責任を負いません。

もくじ

テンプレート概要	1
FreeRTOS要点	2
アプリケーションテンプレ?	3
freertos_generic詳細	4
Base Project作成手順	5
BareMetal → RTOS	6
IoT MCU RTOS etc.	7
テンプレートタスク構成	8
低消費電力アプローチ	9
IoT Baseboard接続	10
参考資料と変更履歴	11