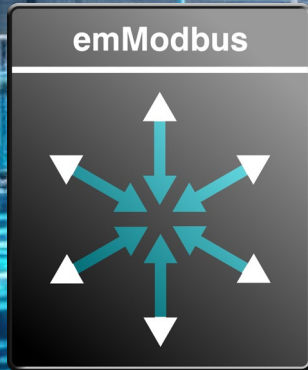


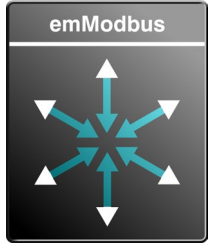


# emModbus Master / Slave

ハードウェア・通信インターフェースに依存性なく利用可能なModbus

Modbus-TCP / ASCII / RTUインターフェース対応

## 組み込みシステムに最適なModbusプロトコルスタック



# emModbus

Modbus Master / Slave対応

## 様々なアプリケーションシナリオで適合できるソリューション

Modbus Master, Slaveそれぞれの製品パッケージを用意。

通信方式は、Modbus Master, Slave共、TCP(UDP), ASCII, RTU対応のサンプルソースコード提供

---

## ハードウェア環境・ソフトウェア環境を問わず利用可能

低リソースで実装可能なため、ハイエンドのMPUからローエンドのMCUまで利用するマイコンハードウェアに依存することなく利用可能

ANSI-Cソースコード提供のためコンパイラやOSの依存性なく利用可能。

---

## 量産ロイヤリティなし、継続コスト不要

開発に対するライセンスのため、量産に対しての継続費用不要

※製品保守は、任意で継続可能です。

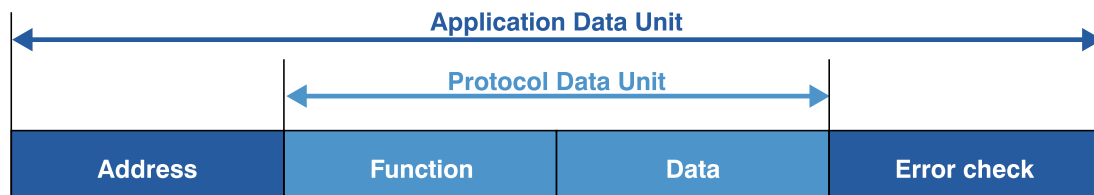
## メッセージフレーム

emModbusは、Modbusで定められている3つの標準プロトコルに対して、ワンパッケージで対応

| プロトコル      | 概要                                                                                  |
|------------|-------------------------------------------------------------------------------------|
| RTU        | オリジナルのModbus規格、RS-232や類するシリアル通信経由で、バイナリーデータを送受信します。                                 |
| ASCII      | RTUと同様にRS-232、類するシリアル通信ですが、RAWバイナリーデータの代わりにASCIIでエンコードしたデータを送受信します。                 |
| Modbus/TCP | Ethernet通信などTCPに基づくヘッダを付与し、バイナリーデータを送受信します。TCPの代替としてUDP通信も利用可能で、Modbus/UDPもサポートします。 |

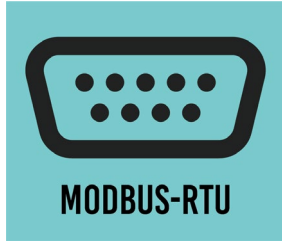
## メッセージフレームフィールド

メッセージフレームは、プロトコルによって異なる方法で処理されますが、RTUとASCIIはどちらも同じ4つの指定されたフィールドで構成されます。フィールド2/3は、プロトコルデータユニットを構成し、Modbus/TCPにも含まれます。

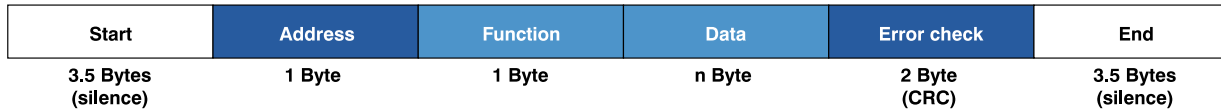




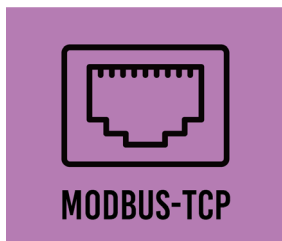
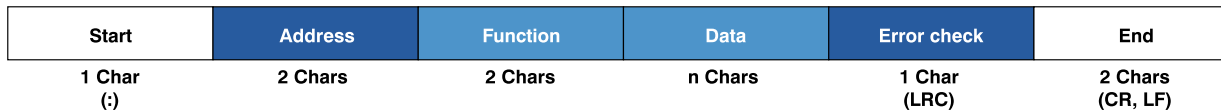
## RTU / ASCII / TCPプロトコル対応



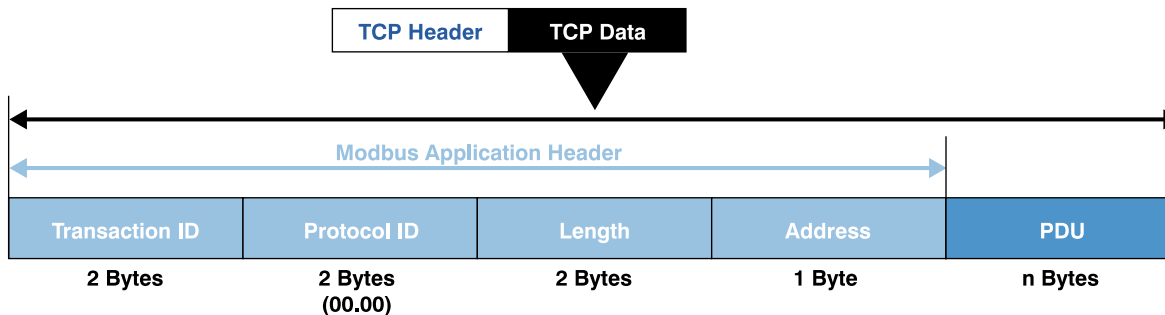
Modbus RTUでは、メッセージに含まれる各バイトはバイナリ データとして送信されます。このモードのメリットは、データ密度が高く、Modbus ASCIIと同じボーレートでデータスループットが向上することです。



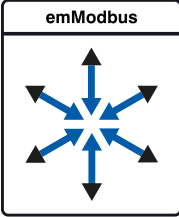
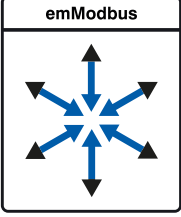
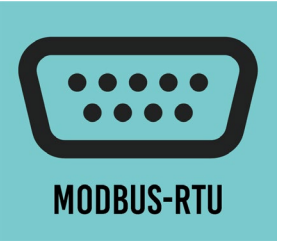
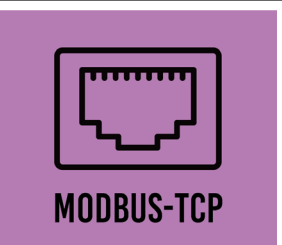
Modbus ASCIIでは、メッセージに含まれる各バイトはエンコードされ、ASCII コードとして送信されます。これにより、エラーを発生させることなく、文字間に最大 1 秒の時間間隔を発生させることができます。



Modbus/TCPでは、「Modbus アプリケーション ヘッダー」と呼ばれる追加のヘッダーがプロトコル データ ユニットの前に配置されます。



Modbus UDPの場合は TCPの代わりにUDPフレームを利用します。

|                                                                                                          |                                                                                                                                    |                                                                                                                          |
|----------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------|
|                                                                                                          | <br><b>emModbus Master</b>                        | <br><b>emModbus Slave</b>             |
| <br><b>MODBUS-RTU</b>   | <b>標準対応</b><br>1:n通信の場合はRTOS推奨<br>SEGGER embOS / FreeRTOS / ITRONなど                                                                | <b>標準対応</b><br>マルチチャンネルSlave対応可能<br>マルチチャンネル実装する場合はRTOS必須                                                                |
| <br><b>MODBUS-ASCII</b> | <b>標準対応</b><br>1:n通信の場合はRTOS推奨<br>SEGGER embOS / FreeRTOS / ITRONなど                                                                | <b>標準対応</b><br>マルチチャンネルSlave対応可能<br>マルチチャンネル実装する場合はRTOS必須                                                                |
| <br><b>MODBUS-TCP</b>  | <b>別途TCP/IPプロトコルスタック</b><br>要BSDソケットインターフェース<br>ITRON仕様TCP/IPスタックに移植可能<br><br>1:n通信の場合はRTOS推奨<br>SEGGER embOS / FreeRTOS / ITRONなど | <b>別途TCP/IPプロトコルスタック</b><br>要BSDソケットインターフェース<br>ITRON仕様TCP/IPスタックに移植可能<br><br>マルチチャンネルSlave対応可能<br>マルチチャンネル実装する場合はRTOS必須 |

## 対応ファンクションコード一覧

emModbusでは、以下のファンクションコードに対応します。ユーザ指定で拡張することも可能です。

| Function Code | 概 要            | Function Code | 概 要                                       |
|---------------|----------------|---------------|-------------------------------------------|
| 1             | コイルの読み取り       | 6             | レジスタの書き込み                                 |
| 2             | ディスクリート入力の読み取り | 8             | Diagnosisファンクション（Subcode:00）※Modbus Slave |
| 3             | HOLDレジスタの読み取り  | 15            | 複数コイルの書き込み                                |
| 4             | 入力レジスタの読み取り    | 16            | 複数レジスタの書き込み                               |
| 5             | コイルの書き込み       | ユーザ指定         | ユーザーコールバックハンドラーで処理します。                    |

## データタイプ

| Data type        | 概 要                                       |
|------------------|-------------------------------------------|
| Coil             | Single bit / read-write / アプリケーションにより変更可能 |
| Discrete Input   | Single bit / read-only / I/Oシステムから提供      |
| Holding Register | 16-bit / read-write / アプリケーションにより変更可能     |
| Input Register   | 16-bit / read-only / I/Oシステムから提供          |

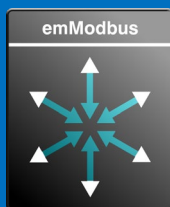
## emModbus 製品パッケージ

emModbusは、アプリケーションに合わせて、Master / Slaveを選択して、導入頂けます。

| Feature                                               | Master | Slave |
|-------------------------------------------------------|--------|-------|
| emModbus Masterソースコード (ASCII, RTU, Modbus/TCPサンプルを含む) | ✓      | ×     |
| emModbus Slaveソースコード (ASCII, RTU, Modbus/TCPサンプルを含む)  | ×      | ✓     |
| Modbus ASCII エンコードフレーム                                | ✓      | ✓     |
| Modbus RTU エンコードフレーム                                  | ✓      | ✓     |
| Modbus/TCP エンコードフレーム (Modbus/UDPサポート)                 | ✓      | ✓     |
| OSカーネル移植レイヤー(Windows / OSレス / embOSのサンプルを含む)          | ✓      | ✓     |
| Modbus Master アプリケーション for Windows (バイナリー)            | ✓      | ✓     |
| Modbus Master アプリケーション for Windows (ソースコード)           | ✓      | ×     |
| Modbus Slave アプリケーション for Windows (バイナリー)             | ✓      | ✓     |
| Modbus Slave アプリケーション for Windows (ソースコード)            | ×      | ✓     |

## コンパクトな実装サイズ

|                 | ROM       | RAM                        |
|-----------------|-----------|----------------------------|
| emModbus Master | 2.5 Kbyte | 30 Byte + 300 Byte (チャンネル) |
| emModbus Slave  | 3.0 Kbyte | 30 Byte + 300 Byte (チャンネル) |



## emModbus 開発・導入支援

- ウェブ公開の製品マニュアル
- サンプルアプリケーション
- 評価ボード無償評価版

使いやすいAPIでアプリケーション開発



APIの利用方法・サンプルコードについては、オンライン製品マニュアルを公開しています。

製品ご導入前にAPI使用方法などについて、確認いただくことが可能です。

# emModbus

CPU independent Modbus  
stack for embedded applications

User Guide & Reference Manual

Document: UM14001  
Software Version: 1.02f  
Revision: 1  
Date: September 24, 2018



A product of SEGGER Microcontroller GmbH

www.segger.com

8

|          |                                         |    |
|----------|-----------------------------------------|----|
| 4        | emModbus API                            | 39 |
| 4.1      | API functions                           | 40 |
| 4.1.1    | Channel specific core functions         | 42 |
| 4.1.1.1  | MB_CHANNEL_Disconnect()                 | 43 |
| 4.1.2    | Master specific core functions          | 44 |
| 4.1.2.1  | MB_MASTER_AddASCIIChannel()             | 45 |
| 4.1.2.2  | MB_MASTER_AddRTUChannel()               | 46 |
| 4.1.2.3  | MB_MASTER_AddRTUChannel()               | 47 |
| 4.1.2.4  | MB_MASTER_DefInit()                     | 48 |
| 4.1.2.5  | MB_MASTER_Init()                        | 49 |
| 4.1.3    | Master instruction set                  | 50 |
| 4.1.3.1  | MB_MASTER_ReadCoil()                    | 51 |
| 4.1.3.2  | MB_MASTER_ReadDI()                      | 52 |
| 4.1.3.3  | MB_MASTER_ReadRI()                      | 53 |
| 4.1.3.4  | MB_MASTER_ReadBC()                      | 54 |
| 4.1.3.5  | MB_MASTER_WriteCoil()                   | 55 |
| 4.1.3.6  | MB_MASTER_WriteCoil()                   | 56 |
| 4.1.3.7  | MB_MASTER_WriteReg()                    | 57 |
| 4.1.3.8  | MB_MASTER_WriteReg()                    | 58 |
| 4.1.4    | Slave specific core function            | 59 |
| 4.1.4.1  | MB_SLAVE_AddASCIIChannel()              | 60 |
| 4.1.4.2  | MB_SLAVE_AddRTUChannel()                | 61 |
| 4.1.4.3  | MB_SLAVE_AddRTUChannel()                | 62 |
| 4.1.4.4  | MB_SLAVE_ConfigureSlaveAddr()           | 63 |
| 4.1.4.5  | MB_SLAVE_DefInit()                      | 64 |
| 4.1.4.6  | MB_SLAVE_Exec()                         | 65 |
| 4.1.4.7  | MB_SLAVE_Init()                         | 66 |
| 4.1.4.8  | MB_SLAVE_PollChannel()                  | 67 |
| 4.1.4.9  | MB_SLAVE_SetCustomFunctionCodeHandler() | 68 |
| 4.1.4.10 | MB_SLAVE_Task()                         | 70 |
| 4.1.5    | Other core functions                    | 71 |
| 4.1.5.1  | MB_ConfigTimerFreq()                    | 72 |
| 4.1.5.2  | MB_OnRx()                               | 73 |
| 4.1.5.3  | MB_OnTx()                               | 74 |
| 4.1.5.4  | MB_TimerTick()                          | 75 |
| 4.1.6    | Helper functions                        | 76 |
| 4.1.6.1  | MB_LoadU16BE()                          | 77 |
| 4.1.6.2  | MB_StoreU16BE()                         | 78 |
| 4.2      | emModbus data structures                | 79 |
| 4.2.1    | Interface configuration structures      | 79 |
| 4.2.1.1  | Structure MB_IFACE_CONFIG_IP            | 79 |
| 4.2.1.2  | Structure MB_IFACE_CONFIG_UART          | 80 |
| 4.2.2    | Interface function structures           | 81 |
| 4.2.2.1  | Structure MB_IFACE_IP_API               | 81 |
| 4.2.2.2  | Structure MB_IFACE_UART_API             | 82 |
| 4.2.3    | Slave structures                        | 83 |
| 4.2.3.1  | Structure MB_SLAVE_API                  | 83 |
| 4.2.3.2  | Structure MB_CUSTOM_FUNC_CODE_PARA      | 84 |
| 4.3      | Error codes                             | 85 |
| 5        | Configuring emModbus                    | 87 |
| 5.1      | Compile-time configuration              | 88 |
| 5.1.1    | Compile-time configuration switches     | 88 |
| 5.1.2    | Debug level                             | 89 |
| 6        | Debugging                               | 91 |
| 6.1      | Message output                          | 92 |
| 6.1.1    | Debug API functions                     | 93 |

emModbus User Guide & Reference Manual © 2014-2018 SEGGER Microcontroller GmbH

46

CHAPTER 4

API functions

### 4.1.2.2 MB\_MASTER\_AddIPChannel()

**Description**  
Adds a master channel that uses the Modbus/TCP or Modbus/UDP protocol via IP.

**Prototype**

```
void MB_MASTER_AddIPChannel( MB_CHANNEL * pChannel,
                             MB_IFACE_CONFIG_IP * pIfcConfig,
                             MB_IP_ADDR * pIPAddr,
                             MB_SLAVE_ADDR * pSlaveAddr,
                             MB_IP_ADDR * pPort);
```

**Parameters**

| Parameter  | Description                                                             |
|------------|-------------------------------------------------------------------------|
| pChannel   | Pointer to element of MB_CHANNEL that is added to linked list.          |
| pIfcConfig | Pointer to element of MB_IFACE_CONFIG_IP                                |
| pIPAddr    | Pointer to element of MB_IP_ADDR; used to read/write from/to interface. |
| Timeout    | Timeout [ms] to wait for answer.                                        |
| SlaveAddr  | Slave addr. to access.                                                  |
| IPAddr     | IP addr. of slave.                                                      |
| Port       | Slave port to connect to.                                               |

**Example**

```
// Static declarations
//
static MB_CHANNEL MBChannel;
static MB_IFACE_CONFIG_IP MBIfcConfig;
static const MB_IP_ADDR IPAddr = {
    .u16 = 192, .u8 = 168, .u16 = 1, .u8 = 1
};
static const MB_SLAVE_ADDR SlaveAddr = {
    .u16 = 192, .u8 = 168, .u16 = 1, .u8 = 1
};
// Code running in the main task
static void MasterTask(void) {
    MB_MASTER_Init(&MBChannel, &MBIfcConfig, &IPAddr, &SlaveAddr, 1);
    MB_MASTER_AddIPChannel(&MBChannel, &MBIfcConfig, &IPAddr, &SlaveAddr, 1);
}
```

emModbus User Guide & Reference Manual © 2014-2018 SEGGER Microcontroller GmbH

37

### 3.1.1 OS\_IP\_MB\_MasterTCP.c

This sample demonstrates emModbus master functionalities using the Modbus/TCP protocol. It opens a channel and tries to establish a TCP connection to a Modbus slave device, which is known to the master by the slave's IP address as defined at the beginning of the file. The master also uses a given port for this connection, which is defined at the beginning of the file (e.g. port 502, the standard port for Modbus communications). When a connection is established, the master repeatedly sends queries to the slave, asking it to perform the function "write coil", and waits for appropriate responses.

### 3.1.2 OS\_IP\_MB\_SlaveTCP.c

This sample demonstrates emModbus slave functionalities using the Modbus/TCP protocol. It opens a channel and waits for incoming TCP connections on a given port, which is known to the slave as defined at the beginning of the file. When an incoming connection from a Modbus master device has been established, the slave reacts to queries it receives from the master. When ordered to write a coil (like the associated Modbus master sample does), the slave will toggle LEDs to signal its new status.

### 3.1.3 OS\_MB\_MasterASCII.c

This emModbus sample demonstrates emModbus master functionalities using ASCII frames. It opens a channel and repeatedly sends queries to a Modbus slave device (specified by its slave ID as defined at the beginning of the file), asking it to perform the function "write coil", and waits for appropriate responses.

### 3.1.4 OS\_MB\_MasterRTU.c

This emModbus sample demonstrates emModbus master functionalities using RTU frames. It opens a channel and repeatedly sends queries to a Modbus slave device (specified by its slave ID as defined at the beginning of the file), asking it to perform the function "write coil", and waits for appropriate responses.

### 3.1.5 OS\_MB\_SlaveASCII.c

This sample demonstrates emModbus slave functionalities using ASCII frames. It opens a channel and waits for incoming queries from a Modbus master device. When ordered to write a coil (like the associated Modbus master sample does), the slave will toggle LEDs to signal its new status.

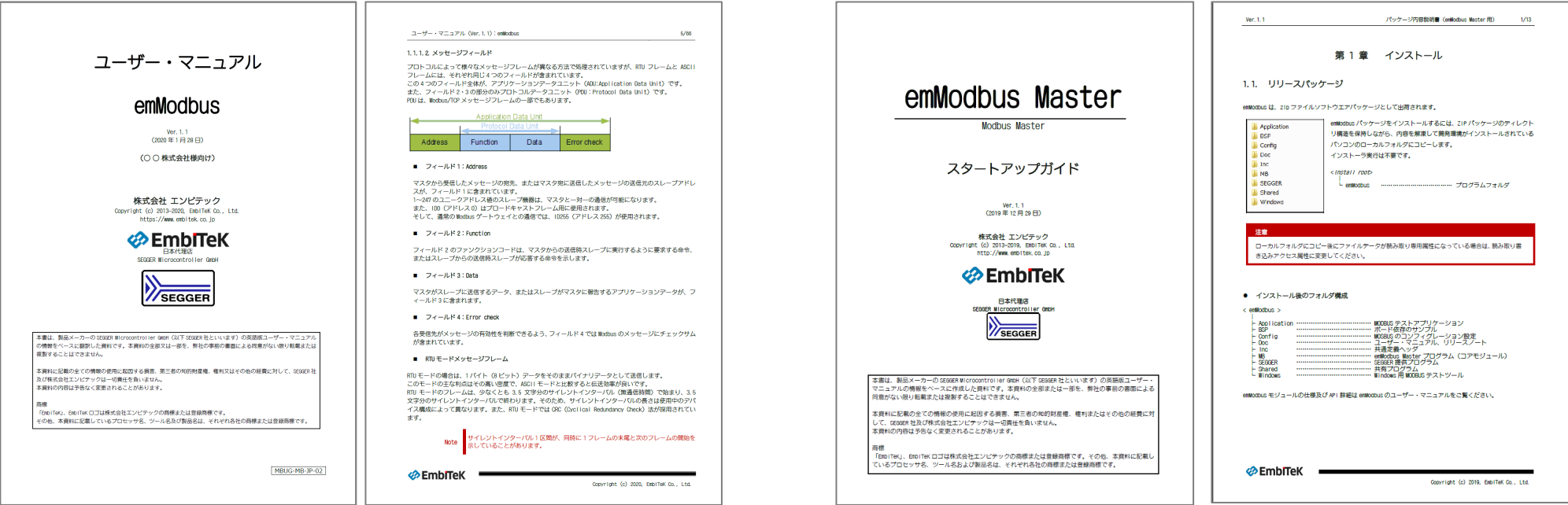
### 3.1.6 OS\_MB\_SlaveRTU.c

This sample demonstrates emModbus slave functionalities using RTU frames. It opens a channel and waits for incoming queries from a Modbus master device. When ordered to write a coil (like the associated Modbus master sample does), the slave will toggle LEDs to signal its new status.

emModbus User Guide & Reference Manual © 2014-2018 SEGGER Microcontroller GmbH

[https://www.segger.com/downloads/emmodbus/UM14001\\_emModbus.pdf](https://www.segger.com/downloads/emmodbus/UM14001_emModbus.pdf)

エンビテックサポート対応あり製品ライセンスで、日本語ドキュメントを提供  
エンビテック作成の日本語マニュアル、スタートアップガイドを提供

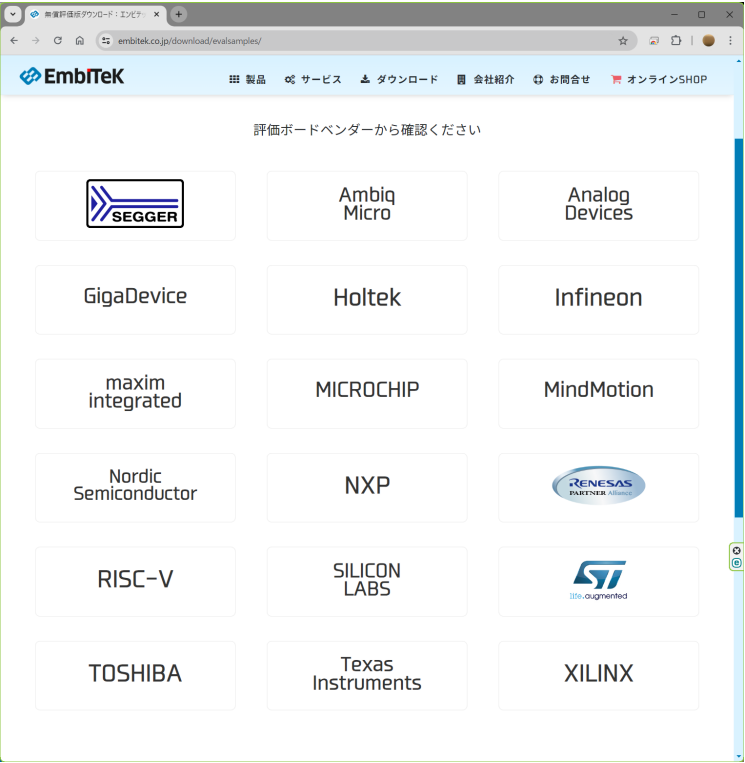


日本語マニュアル

スタートアップガイド

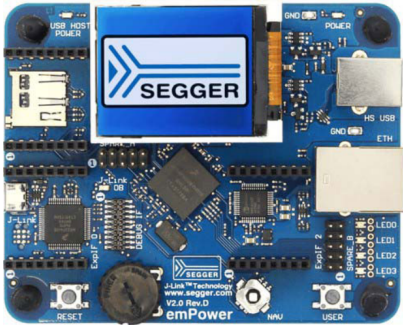
本資料は製品導入後に配布させていただきます。

各種評価ボードで評価サンプルを利用できます。



ウェブからダウンロード可能

### SEGGER emPower



MCU : Kinetis K66 (Arm Cortex-M4) / 180MHz

開発環境 : Embedded Studio [【開発環境無償評価版ダウンロード】](#)

BSP パッケージ内容 :

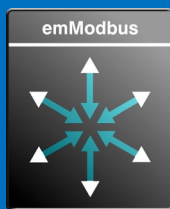
|             |                                                                                                                                                                                                                                                      |
|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| RTOS        | embOS + Profiling                                                                                                                                                                                                                                    |
| 圧縮・解凍       | emCompress-Embed, emCompress-ToGo                                                                                                                                                                                                                    |
| Modbus      | emModbus Master, emModbus Slave                                                                                                                                                                                                                      |
| TCP/IP      | emNet BASE + Web Server, CoAP Server / Client, DHCP Server, (m)DNS/LLMNR/DNS-SD Server, FTP Client, FTP Server, MQTT Client, NetBIOS Name Service, SMTP Client, SNMP Agent, SNTP Client, UPnP, WebSocket, emNet driver for Freescale Kinetis K60/K70 |
| セキュリティ      | emSSH Secure Shell, Secure Copy, emSSL Secure Sockets Layer, emSecure-RSA, emSecure-ECDSA                                                                                                                                                            |
| 暗号・サイファ     | emCrypt PRO                                                                                                                                                                                                                                          |
| IoT Toolkit | HTTP Client, JSON Parser                                                                                                                                                                                                                             |
| GUI         | emWin BASE + AntiAliasing, Bitmap Converter, Font Converter, Memory Devices, Simulation, VNC Server, Widgets, Window Manager, GUIDRV_FlexColor                                                                                                       |
| FileSystem  | emFile BASE + Encryption, FAT, FAT LFN, Journaling, SD/SDHC/SDXC/MMC, NAND, RAMDisk                                                                                                                                                                  |
| USB-Device  | emUSB-Device BASE + Audio, Bulk, CDC, DFU, HID, MSD, MSD-COROM, MTP, Printer Class, IP-over-USB component, VirtualMSD, Video, Target Driver for Freescale Kinetis K60/K70 HighSpeed (EHCI)                                                           |
| USB-Host    | emUSB-Host BASE + Bulk, CDC, FTDI UART, HID, MIDI, MSD, MTP, Printer Class, Freescale Kinetis FullSpeed Driver                                                                                                                                       |

[BSP評価版ダウンロード \(ZIP\)](#)

<https://www.embitek.co.jp/download/evalsamples/>

Windows用デモアプリケーション  
X86用アプリケーションにコンパイルしたemModbusアプリケーション

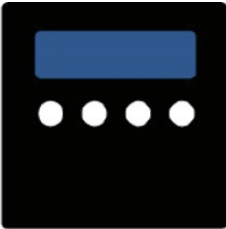
| Software                          |         |              |           |          |
|-----------------------------------|---------|--------------|-----------|----------|
|                                   | Version | Date         | File size |          |
| Windows emModbus demo application | V1.02h  | [2020-05-26] | 895 KB    | DOWNLOAD |



# emModbusライセンスモデル

ニーズに合わせて選択可能なライセンスモデル

永久ライセンス・量産ロイヤリティなしで継続的な費用は必須ではありません。

|                                     | シングルプロダクト                                                                         | プロダクトファミリ<br>(個別提案)                                                                 | シングルデベロッパ<br>(ユーザ)                                                                  | CPU<br>(個別提案)                                                                       |
|-------------------------------------|-----------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|
|                                     |  |  |  |  |
| 開発可能製品数                             | 1製品型番                                                                             | 1製品ファミリ                                                                             | 無制限                                                                                 | 無制限                                                                                 |
| 利用可能開発者数                            | 無制限                                                                               | 無制限                                                                                 | 1名                                                                                  | 無制限                                                                                 |
| CPU                                 | 1CPU型番                                                                            | 1CPU型番                                                                              | 1CPUアーキテクチャ                                                                         | 1CPUアーキテクチャ                                                                         |
| コンパイラ                               | 1種類                                                                               | 1種類                                                                                 | 1種類                                                                                 | 1種類                                                                                 |
| 多数の開発者で1つの製品を開発する。<br>プロジェクト単位で予算計上 |                                                                                   |                                                                                     | 複数の開発プロジェクトで共通利用<br>開発プラットフォーム化に最適                                                  |                                                                                     |



## 開発プロジェクトは無制限／開発者人数に応じたライセンス

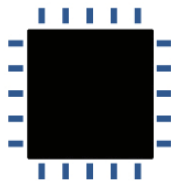
| シングルデベロッパ<br>(ユーザ) | 開発可能製品数 | 利用可能開発者数 | CPU         | コンパイラ |
|--------------------|---------|----------|-------------|-------|
|                    | 無制限     | 1名       | 1CPUアーキテクチャ | 1種類   |



「シングルデベロッパライセンス」は開発プロジェクトに制限されず、無制限に製品開発が可能です。開発者様が複数の開発プロジェクトを担当するなど、多品種開発に最適なライセンスです。

CPUアーキテクチャが同じCPUであれば、製品毎のCPU変更（デバイスメーカー変更）も対応可能です。

| CPU | 開発可能製品数 | 利用可能開発者数 | CPU         | コンパイラ |
|-----|---------|----------|-------------|-------|
|     | 無制限     | 無制限      | 1CPUアーキテクチャ | 1種類   |



「CPUライセンス」は同一CPUアーキテクチャのCPUで複数の開発プロジェクト、開発者の人数に係わらず利用可能です。本ライセンスにより、SEGGER社製RTOS/ミドルウェアを含むソースコードを企業内で、共有ができます。御社内のソフトウェアプラットフォーム化に最適なライセンスです。

本ライセンスは、すべてお客様のご要望に従い都度提案となりますので、必ずしもCPUの制限事項が1CPUアーキテクチャになるわけではなく、ご要望に応じたライセンス提案をさせていただきます。

開発者の人数は無制限（外部協力会社含む）で特定の製品開発に利用可能なライセンス

| シングルプロダクト                                                                                                                                                                 | 開発可能製品数                                                                                                                                                                                                                                   | 利用可能開発者数 | CPU     | コンパイラ |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|---------|-------|
|                                                                                                                                                                           | 1製品型番                                                                                                                                                                                                                                     | 無制限      | 1デバイス型番 | 1種類   |
|                                                                                          | <p>複数の開発者で1つの製品（製品型番）開発が可能です。開発者様が多い大規模開発や品種展開を想定しない製品開発に最適。製品メーカー様へのライセンスで、該当製品開発に係わる開発者は本ライセンスで利用可能です。受託開発で利用検討の場合は、ライセンス契約者として、受託元様での契約をお願いいたします。</p> <p>例）「J-Link BASE」で契約し、「J-Link BASE」を開発する。</p>                                   |          |         |       |
| プロダクトファミリ                                                                                                                                                                 | 開発可能製品数                                                                                                                                                                                                                                   | 利用可能開発者数 | CPU     | コンパイラ |
|                                                                                                                                                                           | 1製品ファミリ                                                                                                                                                                                                                                   | 無制限      | 1デバイス型番 | 1種類   |
| <br> | <p>「プロダクトライセンス」の適用範囲を広げて、1製品シリーズの開発が可能です。開発者様が多い大規模開発で、派生製品開発を行う場合に最適となります。プロダクトファミリの定義は、お客様の要望に応じて、都度SEGGER社と協議の上、ライセンス費用提示となります。</p> <p>例）「J-Linkシリーズ」で契約し、「J-Link BASE」「J-Link PLUS」「J-Link PRO」を開発する。</p> <p>※適用範囲について、適宜ご相談ください。</p> |          |         |       |

# 提供会社

EmbiTeK | SEGGER





## SEGGER Microcontroller GmbH

組み込みシステムで30年以上の経験を持ち、最先端のRTOSおよびソフトウェアライブラリを開発  
ハードウェアツール(開発 / 生産用)とソフトウェアツールをカバーします。

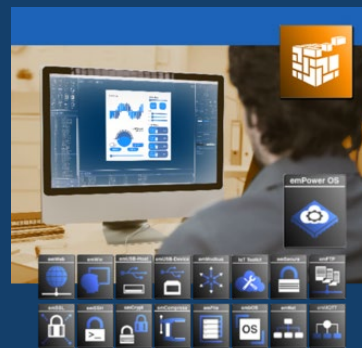
**CEO : Ivo Geilenbruegge**

**設立 : 1992年**

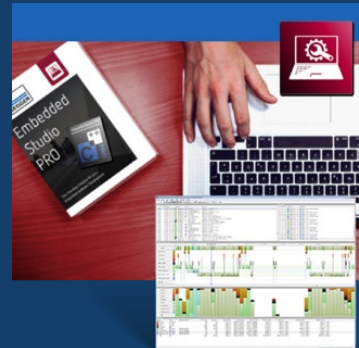
**本社 : モーンハイム・アム・ライン (ドイツ)**

**拠点 : 米国 / 中国**

**30カ国以上に販売代理店を通して展開**



RTOS/ミドルウェア



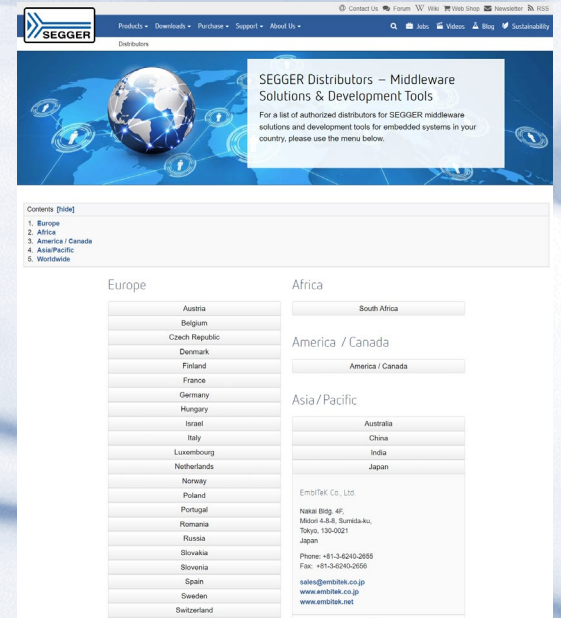
IDE



デバッグツール



書き込みツール



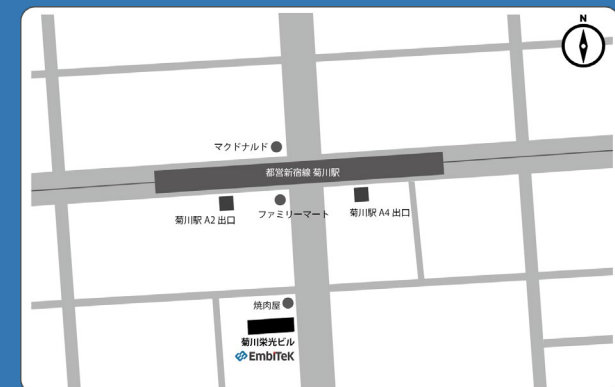
お客様の要件に合わせ、様々なシナリオで適合できる最適なソフトウェア開発環境  
ソフトウェアコンポーネントを提供します。

代表取締役：サントシュ パウル

設立：2007年

本社：東京都墨田区菊川2-3-6 菊川栄光ビル 601

日本国内唯一のSEGGER社製品販売オフィシャルパートナー  
テクニカルサポート／ポーティング受託開発サービスを提供



都営新宿線「菊川駅」徒歩3分



# Arm Cortex/RXソフトウェア開発から量産をサポート

製品開発フローの課題に合わせて対応



デバッガ  
開発ツール

RTOS



MPU 対応



機能安全認証  
IEC61508 SIL3  
IEC62304 class C

|                                                                                                                                                           |                                                                                                                      |                                                                            |                                                                                                 |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------|
| SSL                                                                                                                                                       | 暗号ライブラリ                                                                                                              | セキュリティ認証                                                                   | GUI                                                                                             |
| Modbus                                                                                                                                                    | SSH                                                                                                                  | ブートローダ                                                                     | 圧縮・解凍                                                                                           |
| IoT Toolkit<br>HTTP client<br>JSON Parser                                                                                                                 | MQTT<br>Dropbox Client                                                                                               | USB Host<br>HID<br>MassStorage<br>Printer<br>LAN<br>Audio<br>CCID<br>Video | ファイルシステム<br>NAND<br>SPI/QSPI フラッシュ<br>NOR<br>SD<br>SDHC<br>SDXC<br>MMC<br>eMMC<br>CF<br>USB メモリ |
| TCP/IP<br>IPv4 / IPv6<br>ACP<br>DNS client<br>DNS-SD<br>NetBIOS NS<br>FTP server<br>SNMP Agent<br>PTP OC client<br>Web Socket client<br>Web Socket server | DHCP server<br>ARP<br>mDNS server<br>Loopback<br>CoAP<br>FTP client<br>SNTP client<br>TCP<br>Web server<br>PPP/PPPoE | USB Device<br>HID<br>MTP<br>CDC-NCM<br>RNDIS<br>Printer<br>Audio<br>Bulk   | USB Host<br>MTP<br>CDC<br>FTDI<br>MIDI<br>HUB<br>CP21xx UART                                    |

Arm Cortex / RX CPU

量産書込







製品については、お気軽に以下窓口へお問い合わせください。

TEL : 03-6240-2655  
FAX : 03-6240-2656  
e-mail : sales@embitek.co.jp  
website : <https://www.embitek.co.jp>



**EmbiTeK Online Shop**

<https://www.embitek.shop/>



**YouTube**

<http://www.youtube.com/@embitek>