
PIC®/AVR® IoT - AWSでのデータ送受信 使用者の手引き

序説

著者: Lars Olav Skrebergene, Microchip Technology Inc.



重要: この使用者の手引きは元々 [GitHubのMicrochip IoT開発者の手引き貯蔵庫](#) で公開された一連の指導書の一部です。この貯蔵庫にはAWSでPIC®とAVR®のIoT基板を使うことについてのより多くの指導書と情報があります。

この指導書ではAmazonウェブ サービス(Amazon Web Service®)とMQTTメッセージ規約を使ってクラウドと通信するためにAVR-IoTとPIC-IoTの開発基板がどう構成設定され得るかが示されます。

IoT装置の網がそれらのどれかで釘が押された時に必ずそれらのLEDを点滅するように構成設定される例応用がどう開発され得るかが実演されます。詳細な段階的指示が提供され、必要な時に関連する概念も網羅されます。

この指導書の主な目標は読者に対してAVR-IoTとPIC-IoTの開発基板でそれら自身の応用をどう開発するかを体験することです。

事前要件

読者はこの前の指導書の「[あなたのAWSアカウントへ基板を接続](#)」で記述されるように、それら自身のAWSアカウントと通信するように既に準備されたそれらのPIC-IoTやAVR-IoTの開発基板を持っていると仮定されます。

この指導書を始める前に、そのIoT装置がAWS IoT核へ成功裏にデータを送れていることを確実にしてください。読者はインストールされたMPLAB® X IDEとXC8(AVR-IoT)またはXC16(PIC-IoT)のコンパイラを持つとも仮定されます。

この指導書の最後の「[資料](#)」章でソフトウェアと他の有用な道具と手引きへのリンクが提供されます。



GitHubでコード例を見てください。
貯蔵庫を閲覧するにはクリックしてください。

本書は一般の方々の便宜のため有志により作成されたもので、Microchip社とは無関係であることを御承知ください。しおりの[はじめに]での内容にご注意ください。

目次

序説	1
1. IoT基板のファームウェアに対する簡単な紹介	3
2. 例応用実装	3
2.1. 段階1: GitHubプロジェクトの無変更版での開始	3
2.2. 段階2: クラウドへのMQTTメッセージ送信	3
2.3. 段階3: クラウドからのMQTTメッセージ受信	6
3. 資料	6
4. 改訂履歴	6
Microchipウェブサイト	7
製品変更通知サービス	7
お客様支援	7
Microchipデバイスコード保護機能	7
法的通知	7
商標	8
品質管理システム	8
世界的な販売とサービス	9

1. IoT基板のファームウェアに対する簡単な紹介

PIC-IoTとAVR-IoTの開発基板に予め設定されているファームウェアはGitHubで入手可能で、この指導書で設計される例応用に対する開始点を形成します。違うマイクロ コントローラ系列用のMPLABプロジェクトは以下のここで見つけることができます。

- [PIC-IoT開発基板用GitHub貯蔵庫](#)
- [AVR-IoT開発基板用GitHub貯蔵庫](#)

PICIoT.XとAVRIoT.Xのプロジェクトは暗号、Wi-Fi接続、MQTT通信などを扱う多くの異なるファイルを含みます。この指導書の主な焦点はMPLAB XでSource Files(ソース ファイル)⇒MCC Generated Files(MCC生成ファイル)下に置かれたapplication_manager.cです。このファイルはAWS応用の開発を容易にする多数の有用な高位関数を含みます。

ここはこの指導書に関連するapplication_manager.c内のいくつかの重要な関数の要約です。

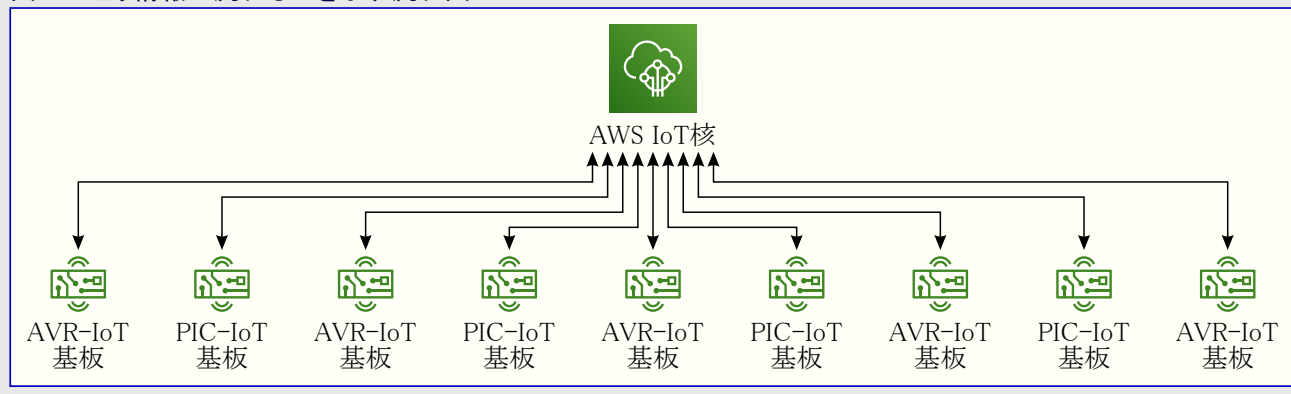
- subscribeToCloud
 - IoT基板で購読されるべきMQTT話題とメッセージがそれらの話題に対して受信される時に走行すべき関数を定義。無変更のPICIoT.XとAVRIoT.Xのプロジェクトでは基板がその装置の影更新MQTT話題に対してだけ購読され、receivedFromCloudがそれらの更新を扱うための関数です。
- receivedFromCloud
 - 無変更のPICIoT.XとAVRIoT.XのプロジェクトでMQTT影更新メッセージが受信された時に走行。この関数はメッセージを分析してその内容に基づいていくつかの活動を実行します。その後影更新が受信されたことを応答するupdateDeviceShadow関数を呼びます。
- sendToCloud
 - 感知器データをMQTTメッセージとしてクラウドへ送るために秒毎に呼ばれます。この関数はIoT基板を使って独自のMQTT話題に独自のMQTTメッセージをどう送るかを学ぶ良い雛形です。

読者はapplication_manager.cを、特に既存の機能の概要を得るためにそれらの関数を覗くことが奨励されます。

2. 例応用実装

この例では独自MQTT話題上で装置がどうメッセージを送って受け取るように構成設定し得るかが実演されます。構成設定された装置のどれかの釦押下が全ての装置にそれらのLEDを閃光させる例応用が実装されます。全ての通信は下の概略図で図解されるようにIoT基板とAWS IoT核間で送られます。装置間直接通信は使われません。

図2-1. どう情報が流れるかを示す流れ図



2.1. 段階1：GitHubプロジェクトの無変更版での開始

この例に対する開始点は使われるデバイスのマイクロ コントローラ系列に適合する以下のようなGitHubプロジェクトの無変更の複製です。

- [PIC-IoT開発基板用GitHub貯蔵庫](#)
- [AVR-IoT開発基板用GitHub貯蔵庫](#)

正しい貯蔵物をダウンロードしてMPLAB XでPICIoT.XやAVRIoT.Xのプロジェクトを開いてください。

2.2. 段階2：クラウドへのMQTTメッセージ送信

最初に行うことは釦が押された時を検出することで、これは割り込みを使って行われます。これに対する手順はAVR-IoTとPIC-IoTの基板に対して多少違います。使う基板に応じて「AVR-IoT基板用手順」または「PIC-IoT基板用手順」に従い、その後は「AVR-IoTとPIC-IoT両基板用手順」を続けてください。

AVR-IoT基板用手順

`application_manager.c`に於いて`application_init`関数で`SYSTEM_Initialize()`呼び出し直後に以下のコードを追加してください。

```
SW0_EnableInterruptForFallingEdge();
PORTF_SW0_SetInterruptHandler(sendButtonPressToCloud);
```

最初の行はAVR-IoT基板のSW0ピンに対する下降端割り込み検出を許可し、この割り込み用の関数処理部がその後に2つ目の行で割り当てられます。興味のある読者はこれらの関数がどう実装されるのかを知るために`pin_manager.c`ファイルを眺めることが奨励されます。

この段階を完了するために「AVR-IoTとPIC-IoT両基板用手順」へ進んでください。

PIC-IoT基板用手順

`pin_manager.c`で以下の編集を実行してください。

1. `PIN_MANAGER_initialize`関数で(RA7に接続される)SW0ピンを許可し、以下のこれらの2行を含めることによってその割り込み要求フラグを解除(0)してください。

```
IOCNAbits.IOCNA7 = 1;    // ピン: RA7
IOCFAbits.IOCA7 = 0;    // ピン: RA7
```

2. SW0ハードウェア割り込み処理部を格納するために`INT_InterruptHandler`変数の下に次のような別の変数を追加してください。

```
void (*SW0_InterruptHandler)(void) = NULL;
```

3. 今作った変数を設定する関数を追加してください(`INT_SetInterruptHandler`関数直後に配置してください)。

```
void SW0_SetInterruptHandler(void (* InterruptHandler)(void))
{
    IEC1bits.IOCIE = 0;           // IOCI割り込み禁止
    SW0_InterruptHandler = InterruptHandler;
    IEC1bits.IOCIE = 1;         // IOCI割り込み許可
}
```

4. `_IOCIInterrupt`割り込み処理ルーチンをSW0ピン押下も扱うように変更してください(割り込み処理ルーチンは`pin_manager.c`の155行近くで見つかります)。SW0ピンはRA7ピンに接続されます。完全に変更された割り込み処理ルーチンが下で提供されます。全体的にそれを複製して`_IOCIInterrupt`を置き換えるか、またはあなたのプロジェクトに入れ子にされた2つ目のif文を追加するかのどちらかを行ってください。

```
void __attribute__(( interrupt, no_auto_psv )) _IOCIInterrupt ( void )
{
    if(IFS1bits.IOCIF == 1)
    {
        // フラグ解除
        IFS1bits.IOCIF = 0;
        if(IOCFAbits.IOCA12 == 1)
        {
            IOCFAbits.IOCA12 = 0;           // ピン(RA12)用フラグ解除
            if(INT_InterruptHandler)
            {
                INT_InterruptHandler();
            }
        }

        // SW0ピン押下処理
        if(IOCFAbits.IOCA7 == 1)
        {
            IOCFAbits.IOCA7 = 0;           // ピン(RA7)用フラグ解除
            if(SW0_InterruptHandler)
            {
                SW0_InterruptHandler();
            }
        }
    }
}
```

pin_manager.hに於いて、他のファイルで利用可能にするために単に追加されたSW0_SetInterruptHandler関数の宣言を、例えばINT_SetInterruptHandler関数の宣言の後に追加してください。

```
void SW0_SetInterruptHandler(void (* InterruptHandler) (void));
```

application_manager.cでapplication_init関数のSYSTEM_Initialize()呼び出し直後にSW0割り込み処理部を設定してください。

```
// 釦押下用割り込み処理部設定
SW0_SetInterruptHandler(sendButtonPressToCloud);
```

AVR-IoTとPIC-IoT両基板用手順

今や、SW0釦が押されると何時でもsendButtonPressToCloud関数が呼ばれます。この関数を実装するのに先立ち、使われるMQTT話題用変数を宣言してください。application_manager.cに以下のコードを追加してください(例えば、mqttSubscribeTopic変数の下に)。

```
char tutorialMqttSubscribeTopic[SUBSCRIBE_TOPIC_SIZE];
```

application_manager.cに以下のコードを追加することによって前述の関数処理部を実装してください。

```
static void sendButtonPressToCloud()
{
    // 有効なクラウド接続があることを保証
    if (shared_networking_params.haveAPConnection)
    {
        static char tutorialPayload[PAYLOAD_SIZE];
        int tutorialLen = 0;

        // MQTT話題設定
        memset((void*)tutorialMqttSubscribeTopic, 0, sizeof(tutorialMqttSubscribeTopic));
        sprintf(tutorialMqttSubscribeTopic, "buttonPresses");

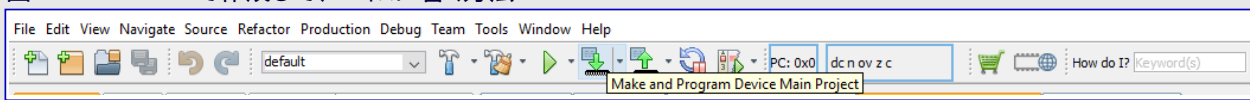
        // 本体構築
        tutorialLen = sprintf(tutorialPayload, "{\"thing_name\":\"%s\"}", cid);

        // クラウドへデータ配信
        CLOUD_publishData((uint8_t*)tutorialMqttSubscribeTopic, (uint8_t*)tutorialPayload,
                          tutorialLen);
    }
}
```

この関数は先に言及したsendToCloud関数に酷似していて、buttonPresses話題にMQTTメッセージを配信します。このメッセージの内容はメッセージを送った物/装置の名前を含むJSONオブジェクトです。

MPLAB Xを使って変更したプロジェクトを構築してそれをデバイスに書いてください。これはMPLAB XツールバーのMake and Program Device Main Project(主プロジェクトを作成してデバイスに書き込み)釦をクリックすることによって行われます。

図2-2. MPLAB Xで作成してデバイスに書く方法



MPLAB X統合開発環境(IDE:Integrated Developer Environment)に不慣れななら、次の手引き、「MPLAB® X IDEとMicrochipツールでの開始に際して」を調べてください。

メッセージが成功裏にAWSへ送られているかの確認

デバイスが成功裏に書かれてしまうと、次の段階はメッセージが成功裏にAWSへ送られているのを確実にすることです。

1. AWS管理操作盤にサインインしてIoT Core service(IoT核サービス)を選んでください。
2. 左手側のメニューでTest(試験)を選んでください。
3. Subscription topic(購読話題)領域でbuttonPressesを入力してください。
4. Subscribe to topic(話題購読)釦をクリックしてください。
5. 基板上でSW0釦を押して釦押下がクラウドで成功裏に登録されることを観察してください。

2.3. 段階3：クラウドからのMQTTメッセージ受信

今や独自の話題へメッセージを送るようにプロジェクトが成功裏に変更され、この話題を購読する方法を見つける時です。

1. 同時に2つまでのMQTT話題購読を許すため、`mqtt_config.h`(Header Files⇒MCC Generated Files⇒config)でNUM_TOPICS_SUBSCRIBEの定義を変更してください。

```
#define NUM_TOPICS_SUBSCRIBE 2
```

2. `buttonPresses`話題購読を含めるように`application_manager.c`で`subscribeToCloud`関数を編集してください。完全に変更した関数が下で提供されます。全体的にそれを複製して`subscribeToCloud`を置き換えるか、またはあなたのMPLAB Xプロジェクトで下のコードの最後の2行を追加するかのどちらかを行ってください。

```
static void subscribeToCloud(void)
{
    sprintf(mqttSubscribeTopic, "$aws/things/%s/shadow/update/delta", cid);
    QCLOUD_registerSubscription((uint8_t*)mqttSubscribeTopic, receivedFromCloud);
    sprintf(tutorialMqttSubscribeTopic, "buttonPresses");
    CLOUD_registerSubscription((uint8_t*)tutorialMqttSubscribeTopic,
                               receiveButtonPressFromCloud);
}
```

- `CLOUD_registerSubscription`関数の2つ目の要素は指定した話題に対するメッセージが受信された時にどの関数が走るかを指示する処理部です。従って、`receiveButtonPressFromCloud`関数は受信したどのメッセージも処理するように実装されなければなりません。
3. メッセージが受信された時に装置のLEDが2度点滅するように`application_manager.c`(で`subscribeToCloud`関数の上の何処か)に以下の関数定義を追加してください。

```
static void receiveButtonPressFromCloud(uint8_t *topic, uint8_t *payload)
{
    LED_test();
    LED_test();
}
```

4. MPLAB Xでプロジェクトを構築してデバイスを書いてください。複数のAVR-IoTやPIC-IoTの装置が利用可能なら、同じプロジェクトを使ってそれらのいくつかを書いてみてください。

最初に全ての装置がAWSアカウントで使うように準備されなければならないことを覚えて置いてください。例えAVR-IoTとPIC-IoTの装置が同時にAWSに接続され、MQTT上で互いに通信することもできるとしても、この指導書で使われるGitHubプロジェクトはAVR-IoT装置またはPIC-IoT装置のどちらかだけに適合することにも注意してください。2つの異なるデバイス系列のデバイスを共に使うには、GitHub上のAVR-IoTとPIC-IoTの貯蔵庫に対してこの指導書を個別に完了してデバイスを適合するファームウェアで書くことが必要です。

装置は今や正しく構成設定されたでしょう。構成設定したIoTキットのどれかでSW0釦が押された場合、構成設定した全てのIoTキット上のLEDが2度点滅するでしょう。これが行われない場合、この指導書に正しく従っていることと、装置が正しい状態にされていることを確実にしてください。

3. 資料

- [PIC-IoT WA開発基板製品頁](#)
- [AVR-IoT WA開発基板製品頁](#)
- [MPLAB® XとMicrochipツールでの開始に際して](#)
- [AWS IoT開発者の手引き](#)
- [IoT準備ツール](#)
- IoT基板の予め設定されたファームウェアを持つGitHub貯蔵庫
 - [PIC-IoT開発基板用](#)
 - [AVR-IoT開発基板用](#)

4. 改訂履歴

資料改訂	日付	注釈
A	2020年8月	初版文書公開

Microchipウェブ サイト

Microchipはwww.microchip.com/で当社のウェブ サイト経由でのオンライン支援を提供します。このウェブ サイトはお客様がファイルや情報を容易に利用可能にするのに使われます。利用可能な情報のいくつかは以下を含みます。

- **製品支援** – データシートと障害情報、応用記述と試供プログラム、設計資源、使用者の手引きとハードウェア支援資料、最新ソフトウェア配布と保管されたソフトウェア
- **一般的な技術支援** – 良くある質問(FAQ)、技術支援要求、オンライン検討グループ、Microchip設計協力課程会員一覧
- **Microchipの事業** – 製品選択器と注文の手引き、最新Microchip報道発表、セミナーとイベントの一覧、Microchip営業所の一覧、代理店と代表する工場

製品変更通知サービス

Microchipの製品変更通知サービスはMicrochip製品を最新に保つのに役立ちます。加入者は指定した製品系統や興味のある開発ツールに関連する変更、更新、改訂、障害情報がある場合に必ず電子メール通知を受け取ります。

登録するにはwww.microchip.com/pcnへ行って登録指示に従ってください。

お客様支援

Microchip製品の使用者は以下のいくつかのチャネルを通して支援を受け取ることができます。

- 代理店または販売会社
- 最寄りの営業所
- 組み込み解決技術者(ESE:Embedded Solutions Engineer)
- 技術支援

お客様は支援に関してこれらの代理店、販売会社、またはESEに連絡を取るべきです。最寄りの営業所もお客様の手助けに利用できます。営業所と位置の一覧はこの資料の後ろに含まれます。

技術支援はwww.microchip.com/supportでのウェブ サイトを通して利用できます。

Microchipデバイスコード保護機能

Microchipデバイスでの以下のコード保護機能の詳細に注意してください。

- Microchip製品はそれら特定のMicrochipデータシートに含まれる仕様に合致します。
- Microchipは意図した方法と通常条件下で使われる時に、その製品系統が安全であると考えます。
- Microchipデバイスのコード保護機能を破ろうとする試みに使われる不正でおそらく違法な方法があります。当社はこれらの方法がMicrochipのデータシートに含まれた動作仕様外の方法でMicrochip製品を使うことが必要とされると確信しています。これらのコード保護機能を破ろうとする試みは、おそらく、Microchipの知的財産権に違反することなく達成することはできません。
- Microchipはそのコードの完全性について心配されている何れのお客様とも共に働きたいと思えます。
- Microchipや他のどの半導体製造業者もそのコードの安全を保証することはできません。コード保護は製品が”破ることができない”ことを当社が保証すると言うことを意味しません。コード保護は常に進化しています。Microchipは当社製品のコード保護機能を継続的に改善することを約束します。Microchipのコード保護機能を破る試みはデジタル ミレニアム著作権法に違反するかもしれません。そのような行為があなたのソフトウェアや他の著作物に不正なアクセスを許す場合、その法律下の救済のために訴権を持つかもしれません。

法的通知

この刊行物に含まれる情報はMicrochip製品を使って設計する唯一の目的のために提供されます。デバイス応用などに関する情報は皆さまの便宜のためにだけ提供され、更新によって取り換えられるかもしれません。皆さまの応用が皆さまの仕様に合致するのを保証するのは皆さまの責任です。

この情報はMicrochipによって「現状そのまま」で提供されます。Microchipは非侵害、商品性、特定目的に対する適合性の何れの黙示的保証やその条件、品質、性能に関する保証を含め、明示的にも黙示的にもその情報に関連して書面または表記された書面または黙示の如何なる表明や保証もしません。

如何なる場合においても、Microchipは情報またはその使用に関連するあらゆる種類の間接的、特別的、懲罰的、偶発的または結果的な損失、損害、費用または経費に対して責任を負わないものとします。法律で認められている最大限の範囲で、情報またはその使用に関連する全ての請求に対するMicrochipの全責任は、もしあれば、情報のためにMicrochipへ直接支払った料金を超えないものとします。生命維持や安全応用でのMicrochipデバイスの使用は完全に購入者の危険性で、購入者はそのような使用に起因する全ての損害、請求、訴訟、費用からMicrochipを擁護し、補償し、免責することに同意します。他に言及されない限り、Microchipのどの知的財産権下でも暗黙的または違う方法で許認可は譲渡されません。

商標

Microchipの名前とロゴ、Mchip、Adapteck、AnyRate、AVR、AVR、AVR Freaks、BesTime、BitCloud、chipKIT、chipKIT、CryptoMemory、CryptoRF、dsPIC、FlashFlex、flexPWR、HELDO、IGLOO、JukeBlox、KeeLoq、Kleer、LANCheck、LinkMD、maXStylus、maXTouch、MediaLB、megaAVR、Microsemi、Microsemi、MOST、MOST、MPLAB、OptoLyzer、PacTime、PIC、picoPower、PICSTART、PIC32、PolarFire、Prochip Designer、QTouch、SAM-BA、SenGenuity、SpyNIC、SST、SST、SuperFlash、Symmetricom、SyncServer、Tachyon、TempTracker、TimeSource、tinyAVR、UNI/O、Vectron、XMEGAは米国と他の国に於けるMicrochip Technology Incorporatedの登録商標です。

APT、ClockWorks、The Embedded Control Solutions Company、EtherSynch、FlashTec、Hyper Speed Control、HyperLight Load、IntelliMOS、Liberio、motorBench、mTouch、Powermite 3、Precision Edge、ProASIC、ProASIC Plus、ProASIC Plus、Quiet-Wire、SmartFusion、SyncWorld、Temux、TimeCesium、TimeHub、TimePictra、TimeProvider、Vite、WinPath、ZLは米国に於けるMicrochip Technology Incorporatedの登録商標です。

Adjacent Key Suppression、AKS、Analog-for-the-Digital Age、Any Capacitor、AnyIn、AnyOut、BlueSky、BodyCom、CodeGuard、CryptoAuthentication、CryptoCompanion、CryptoController、dsPICDEM、dsPICDEM.net、Dynamic Average Matching、DAM、ECAN、EtherGREEN、In-Circuit Serial Programming、ICSP、INICnet、Inter-Chip Connectivity、JitterBlocker、KleerNet、KleerNet、memBrain、Mindi、MiWi、MPASM、MPF、MPLAB Certified、MPLAB、MPLINK、MultiTRAK、NetDetach、Omniscient Code Generation、PICDEM、PICDEM.net、PICkit、PICtail、PowerSmart、PureSilicon、QMatrix、REALICE、Ripple Blocker、SAM-ICE、Serial Quad I/O、SMART-I.S.、SQI、SuperSwitcher、SuperSwitcher II、Total Endurance、TSHARC、USBCheck、VariSense、View Sense、WiperLock、Wireless DNA、ZENAは米国と他の国に於けるMicrochip Technology Incorporatedの商標です。

SQTPは米国に於けるMicrochip Technology Incorporatedの役務標章です。

Adapteck、Frequency on Demand、Silicon Storage Technology、Symmcomは他の国に於けるMicrochip Technology Inc.の登録商標です。

GestICは他の国に於けるMicrochip Technology Inc.の子会社であるMicrochip Technology Germany II GmbH & Co. KGの登録商標です。

ここで言及した以外の全ての商標はそれら各々の会社の所有物です。

© 2020年、Microchip Technology Incorporated、米国印刷、不許複製

品質管理システム

Microchipの品質管理システムに関する情報についてはwww.microchip.com/qualityを訪ねてください。

日本語© HERO 2020.

本使用者の手引きはMicrochipのPIC®/AVR® IoT - AWSでのデータ送受信使用者の手引き(DS50003021A-2020年8月)の翻訳日本語版です。日本語では不自然となる重複する形容表現は省略されている場合があります。日本語では難解となる表現は大幅に意識されている部分もあります。必要に応じて一部加筆されています。頁割の変更により、原本より頁数が少なくなっています。

必要と思われる部分には()内に英語表記や略称などを残す形で表記しています。

青字の部分はリンクとなっています。一般的に赤字の0,1は論理0,1を表します。その他の赤字は重要な部分を表します。



MICROCHIP

世界的な販売とサービス

米国	亜細亜/太平洋	亜細亜/太平洋	欧州
本社 2355 West Chandler Blvd. Chandler, AZ 85224-6199 Tel: 480-792-7200 Fax: 480-792-7277 技術支援: www.microchip.com/support ウェブアドレス: www.microchip.com アトランタ Duluth, GA Tel: 678-957-9614 Fax: 678-957-1455 オースチン TX Tel: 512-257-3370 ボストン Westborough, MA Tel: 774-760-0087 Fax: 774-760-0088 シカゴ Itasca, IL Tel: 630-285-0071 Fax: 630-285-0075 ダラス Addison, TX Tel: 972-818-7423 Fax: 972-818-2924 デトロイト Novi, MI Tel: 248-848-4000 ヒューストン TX Tel: 281-894-5983 インディアナポリス Noblesville, IN Tel: 317-773-8323 Fax: 317-773-5453 Tel: 317-536-2380 ロサンゼルス Mission Viejo, CA Tel: 949-462-9523 Fax: 949-462-9608 Tel: 951-273-7800 ローリー NC Tel: 919-844-7510 ニューヨーク NY Tel: 631-435-6000 サンホセ CA Tel: 408-735-9110 Tel: 408-436-4270 カナダ - トロント Tel: 905-695-1980 Fax: 905-695-2078	オーストラリア - シドニー Tel: 61-2-9868-6733 中国 - 北京 Tel: 86-10-8569-7000 中国 - 成都 Tel: 86-28-8665-5511 中国 - 重慶 Tel: 86-23-8980-9588 中国 - 東莞 Tel: 86-769-8702-9880 中国 - 広州 Tel: 86-20-8755-8029 中国 - 杭州 Tel: 86-571-8792-8115 中国 - 香港特別行政区 Tel: 852-2943-5100 中国 - 南京 Tel: 86-25-8473-2460 中国 - 青島 Tel: 86-532-8502-7355 中国 - 上海 Tel: 86-21-3326-8000 中国 - 瀋陽 Tel: 86-24-2334-2829 中国 - 深圳 Tel: 86-755-8864-2200 中国 - 蘇州 Tel: 86-186-6233-1526 中国 - 武漢 Tel: 86-27-5980-5300 中国 - 西安 Tel: 86-29-8833-7252 中国 - 廈門 Tel: 86-592-2388138 中国 - 珠海 Tel: 86-756-3210040	インド - ハンガロール Tel: 91-80-3090-4444 インド - ニューデリー Tel: 91-11-4160-8631 インド - フネー Tel: 91-20-4121-0141 日本 - 大阪 Tel: 81-6-6152-7160 日本 - 東京 Tel: 81-3-6880-3770 韓国 - 大邱 Tel: 82-53-744-4301 韓国 - ソウル Tel: 82-2-554-7200 マレーシア - クアラルンプール Tel: 60-3-7651-7906 マレーシア - ペナン Tel: 60-4-227-8870 フィリピン - マニラ Tel: 63-2-634-9065 シンガポール Tel: 65-6334-8870 台湾 - 新竹 Tel: 886-3-577-8366 台湾 - 高雄 Tel: 886-7-213-7830 台湾 - 台北 Tel: 886-2-2508-8600 タイ - バンコク Tel: 66-2-694-1351 ベトナム - ホーチミン Tel: 84-28-5448-2100	オーストラリア - ウェルズ Tel: 43-7242-2244-39 Fax: 43-7242-2244-393 デンマーク - コペンハーゲン Tel: 45-4485-5910 Fax: 45-4485-2829 フィンランド - エスポー Tel: 358-9-4520-820 フランス - パリ Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79 ドイツ - ガルヒング Tel: 49-8931-9700 ドイツ - ハーン Tel: 49-2129-3766400 ドイツ - ハイムブロン Tel: 49-7131-72400 ドイツ - カールスルーエ Tel: 49-721-625370 ドイツ - ミュンヘン Tel: 49-89-627-144-0 Fax: 49-89-627-144-44 ドイツ - ローゼンハイム Tel: 49-8031-354-560 イスラエル - ラーナナ Tel: 972-9-744-7705 イタリア - ミラノ Tel: 39-0331-742611 Fax: 39-0331-466781 イタリア - ハドバ Tel: 39-049-7625286 オランダ - デルフト Tel: 31-416-690399 Fax: 31-416-690340 ノルウェー - トロンハイム Tel: 47-72884388 ポーランド - ワルシャワ Tel: 48-22-3325737 ルーマニア - ブカレスト Tel: 40-21-407-87-50 スペイン - マドリッド Tel: 34-91-708-08-90 Fax: 34-91-708-08-91 スウェーデン - イェテボリ Tel: 46-31-704-60-40 スウェーデン - ストックホルム Tel: 46-8-5090-4654 イギリス - ウォーキングム Tel: 44-118-921-5800 Fax: 44-118-921-5820