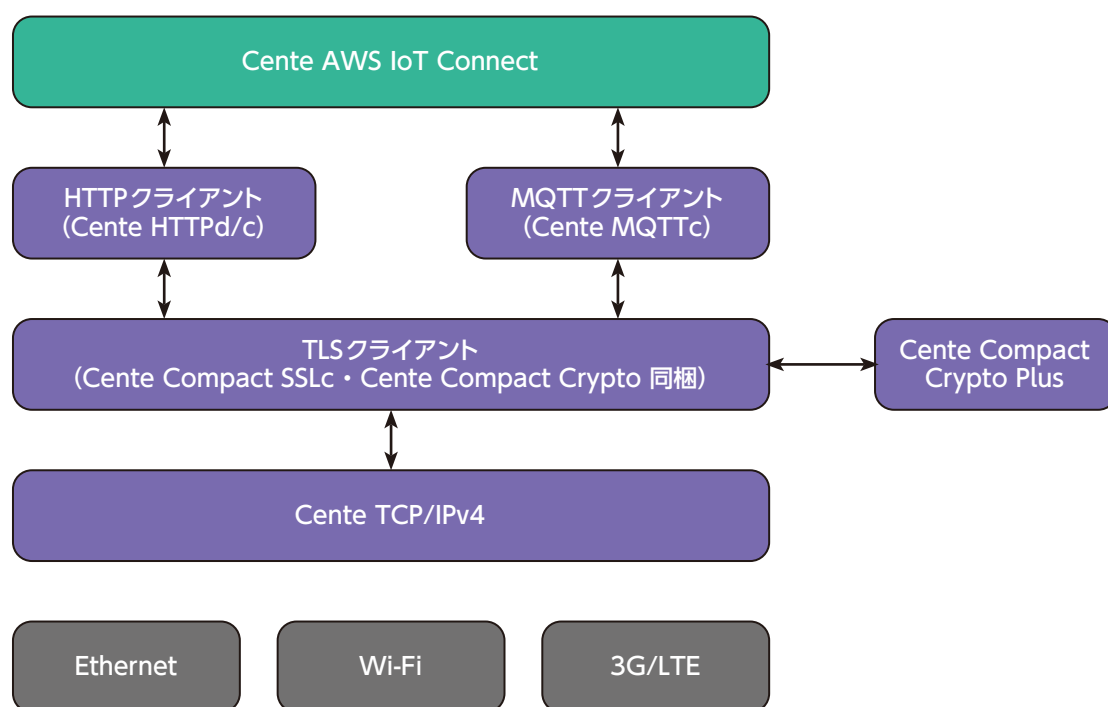


概要

Cente AWS IoT Connectは必要なCenteミドルウェアを利用しAmazon AWS IoT Coreに組み込み機器からアクセスする際に必要なソースコード、実装手順、接続手順をまとめたアプリケーションノートです。

AWS IoT Coreとの接続手順、各種ミドルウェアの組み込み手順、アプリケーションサンプルを網羅しており、導入後直ちにAWS IoT Coreとのセキュアな通信路を利用したシステム開発に集中することが可能です。

概念図



製品構成

●ドキュメント 実装編

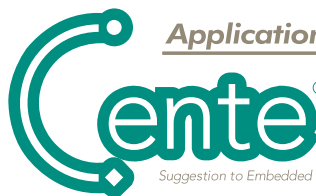
- Cente TCP/IPv4 上へのプロジェクト追加手順
- Cente Compact SSLc のAPI 設定
- Cente MQTTc のAPI 設定

●ソースプログラム

- 試験仕様書付き接続サンプルプログラム
- デバッグコード ※コンパイル時に取り外し可能
- AWS IoT 用プログラム差分

●ドキュメント 接続手順編

- AWS IoT のMQTT サーバ構築方法の手順説明
 - ・ AWS マネージメントコンソールへのログイン
 - ・ AWS IoT Core へのアクセス
 - ・ MQTT サーバ構築
 - ・ クライアントURL 取得
- Cente MQTTc、Cente HTTPd/c(HTTPc)からの接続方法
 - ・ AWS IoT ポリシーの設定
 - ・ 必要情報の設定
 - ・ AWS IoT Core からの動作確認手順
 - ・ AWS IoT のMQTT の制約について
- 証明書、秘密鍵の形式について
- フリープロビジョニングに対応



Application Package

AWS IoT Connect

powered by aws

■ AWS IoT Connectを利用するにあたり

組み込み機器にてAmazon AWS IoTとのデータ通信を実現するためにはCente AWS IoT Connect以外に以下のパッケージが必要になります。

Cente TCP/IPv4
Cente Compact SSLc Ver1.40以降
Cente Compact Crypto Plus※
Cente MQTTC Ver1.10 以降
または Cente HTTPd/c Ver1.70 以降

■ ハードウェアアクセラレータの使用について

Cente Compact SSLcはソフトウェア暗号・復号アルゴリズムの他、ハードウェアアクセラレータによる高速暗号・認証にも対応可能です。

Cente Compact SSLcは様々なハードウェアに対応可能ですが、ハードウェアの指定や条件によって動作環境が異なりますので、別途ご相談の上でのご提供とさせていただきます。

■ 制限事項など

AWS IoTに必要なTLSのクライアント認証は組み込み機器には重い処理で、CPUの性能によっては数十秒程度の時間が必要です。通信接続・切断を頻繁に行うシステムの場合、十分な考慮をお願いします。

■ API関数一覧

● MQTT

mqtt_connect	AWS IoT Coreと接続
mqtt_disconnect	AWS IoT Coreから切断
mqtt_publish	データ送信
mqtt_subscribe	受信データの指定
mqtt_unsubscribe	受信データ指定の解除
mqtt_set_callbackfunc	データ受信、通信エラーコールバック関数設定
mqtt_read_rcv_publish	受信データ取り出し

■ 対応暗号スイート

ECDHE-ECDSA-AES128-GCM-SHA256 (推奨)
ECDHE-RSA-AES128-GCM-SHA256 (推奨)
ECDHE-ECDSA-AES128-SHA256
ECDHE-RSA-AES128-SHA256
ECDHE-ECDSA-AES128-SHA
ECDHE-RSA-AES128-SHA
ECDHE-ECDSA-AES256-GCM-SHA384
ECDHE-RSA-AES256-GCM-SHA384
ECDHE-ECDSA-AES256-SHA384
ECDHE-RSA-AES256-SHA384
ECDHE-RSA-AES256-SHA
ECDHE-ECDSA-AES256-SHA
AES128-GCM-SHA256
AES128-SHA256
AES128-SHA
AES256-GCM-SHA384
AES256-SHA256
AES256-SHA

※ ... [Cente Compact Crypto Plus]を利用した場合
暗号スイートの表記、順番はAWS IoTのサイトに記載されている内容に準じています

● HTTP

httpc_init_reqheader()	リクエストヘッダを初期化する
httpc_init_rspheader()	レスポンスヘッダを初期化する
httpc_create_socket_buffer()	HTTP通信ソケットバッファを取得する
httpc_post()	HTTP POSTリクエストを送信する
httpc_post_with_sock()	HTTP POSTリクエストを送信する (連続呼び出し可能)
httpc_disconnect()	HTTPサーバとTCP切断を行う
httpc_delete_socket_buffer()	HTTP通信ソケットバッファを開放する

● μTRONはMicro Industrial The Realtime Operating system Nucleusの略称です。

● CenteはNXTech株式会社とDMG MORI Digital 株式会社の登録商標です。 ● その他の製品名は各メーカーの商標又は登録商標です。

【開発・製造・販売】

NXTech株式会社 デジタル事業部 IoTソリューション部

〒190-0022 東京都立川市錦町1-8-7 立川錦町ビル8F
TEL.042-523-1177 FAX.042-523-7070

DMG MORI Digital株式会社

〒004-0015 札幌市厚別区下野幌テクノパーク1-1-14
TEL.011-807-6666

● 問い合わせ先：詳しくはサイトをご覧ください

E-mail: sales@cente.jp
TEL.042-523-1177

技術セミナー開催中!
cente.jp

【販売代理店】