

Googleアシスタントでの音声制御

序説

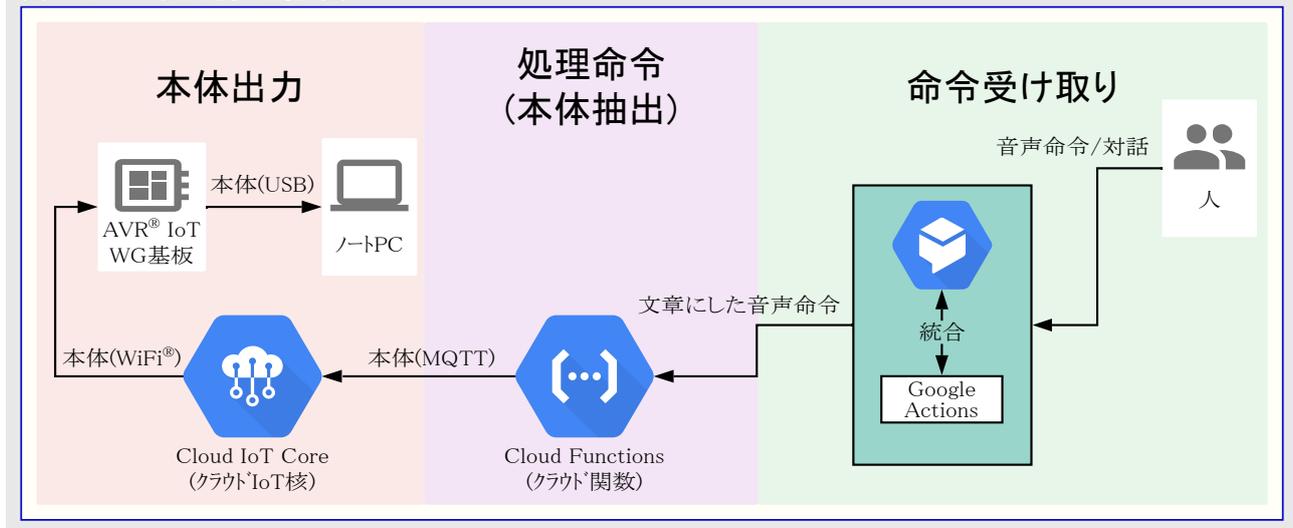
著者: Johan Lofstad, Microchip Technology Inc.

この使用者の手引きはGoogleクラウド基盤(GCP:Google Cloud Platform)を使ってどうGoogle HomeアシスタントをIoTプロジェクトに統合することができるかを記述します。特に、番号を出力するのにAVR[®]-IoT WG基板を使う”番号出力”例。”制御卓に123を出力”のような音声命令が与えられると、IoT基板は命令行に番号を出力します。網羅される話題は次のとおりです。

- クラウドへのAVR-IoT基板接続
- DialogflowとGoogle Actionsを使ってGoogleクラウドプロジェクトへGoogle Homeアシスタント対応装置を統合
- AVR-IoT基板に”print x”メッセージを送るための音声命令の追加

図1はメッセージ出力例の流れ図を示します。Google Homeアシスタント対応装置に話すことによって人は音声命令を始めます。音声はDialogflowとGoogle Actionsを通して処理され、音声から命令を理解します。その命令はその後に文章としてCloud Functions(クラウド関数)へ転送されます。クラウド関数は要求を処理してCloud IoT Core(クラウドIoT核)を使ってAVR-IoT基板にメッセージを転送します。この基板がメッセージを受け取ると、それはUSBを通して接続されたノートPCへ即座に送られます。

図1. メッセージ出力例の流れ図



助言: IoT基板用の例ソースコードはAtmel START:https://start.atmel.com/#example/Atmel%3AAVR_IoT_WG_Sensor_Node_With_Voice_Control%3A1.0.0%3A%3AApplication%3AAVR_IoT_WG_Sensor_Node_With_Voice_Control%3Aで見つけることができます。

Googleクラウド基盤用の例ソースコードはGitHub:<https://github.com/microchip-pic-avr-solutions/avr-iot-wg-board-voice-control-cloud>で見つけることができます。

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. クラウド構成設定	3
1.1. IoT核構成設定	3
1.1.1. 装置を登録部へ追加	3
1.2. クラウド関数設定	4
1.3. DialogflowとGoogle Actions	7
1.3.1. インテント作成	7
1.3.2. 対話の試験と確認	9
1.3.3. Googleアシスタント対応装置の使い方	10
2. AVR-IoT WG基板追加	10
2.1. メッセージ処理	11
3. 改訂履歴	12
Microchipウェブサイト	13
製品変更通知サービス	13
お客様支援	13
Microchipデバイスコード保護機能	13
法的通知	13
商標	14
品質管理システム	14
世界的な販売とサービス	15

1. クラウド構成設定

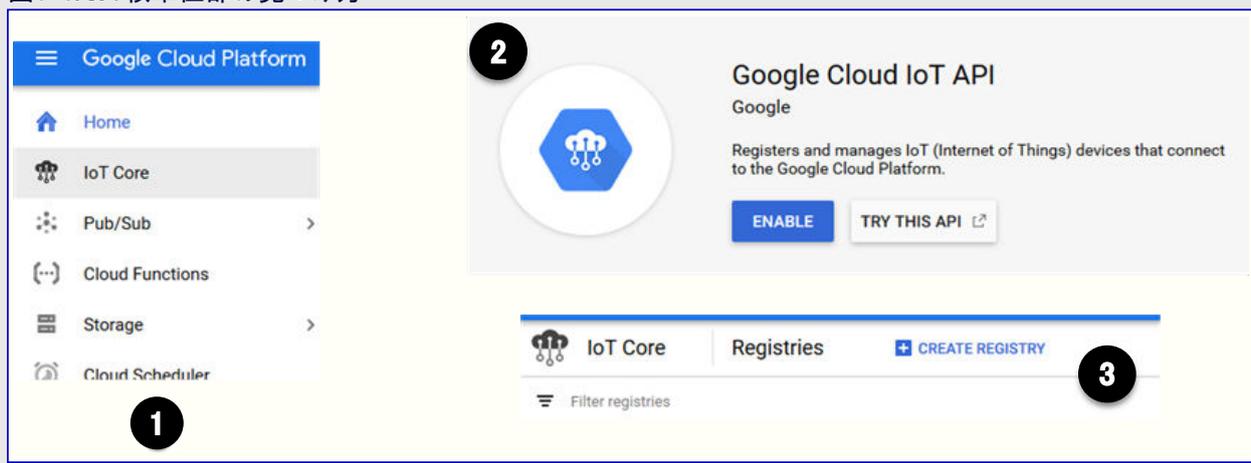
Googleアシスタント装置を統合するのにGoogleクラウド基盤アカウントとプロジェクトが必要とされます。<https://cloud.google.com>へ誘導してアカウントを作成してください。プロジェクトの作成を促される時に”Vloice to AVR”のように関連する何かで名付けてください。プロジェクトが有効になるのに数分かかるかもしれません。

1.1. IoT核構成設定

IoT核単位部はIoT装置との全ての通信を扱うように設計されています。全ての装置は固有IDと認証資格と共にIoT核で登録されます。要するに、これはIoT装置とクラウドの残りの間の交換器として扱います。

IoT核単位部はクラウド制御卓の左手側のメニューを通して開くことができます。クラウドプロジェクトに単位部を追加するには”Enable(許可)”をクリックしてください。IoT核が追加されると、Registries(登録部)頁が現れるべきです。この手続きの画面例については図1-1をご覧ください。

図1-1. IoT核単位部の見つけ方



IoT核単位部に接続するには装置が登録部に追加されなければなりません。登録部はクラウドと通信することができる装置の組です。新しい登録部を作成するにはCreate Registry(登録部作成)鈕をクリックしてください。いくつかの必要とされる領域があります。表1-1の”入力”列に従って登録部を構成設定してください。いくつかの領域は”Show Advanced Options(高度な任意選択を表示)”をクリックする前に現れないかもしれません。

表1-1. IoT核作成登録部領域

名前	入力	説明
Registry ID	voice-devices	登録部を識別する恒久的なID
Region	適用地域	データが格納される地理的な地域
Protocol	<input checked="" type="checkbox"/> MQTT <input type="checkbox"/> HTTP	どの通信規約が登録部を支援するか?。MQTTとHTTPの両方が支援されます。
Cloud Pub/Sub topics	引き落としメニューを選んでCreate a topic(話題作成)を選んでください。話題名voice-upstreamを入力してください。残りを既定のままとして”Create topic(話題作成)”を押してください。	既定の遠隔測定の話題(トピック)はMQTT話題で、これは装置からの全メッセージが配送されます。
Device state topic (任意選択)	無変化のままにしてください。	装置によって配給される全ての状態事象がそれらに送られます。この例では使われません。
Stackdriver Logging	None(なし)	この例では使われません。

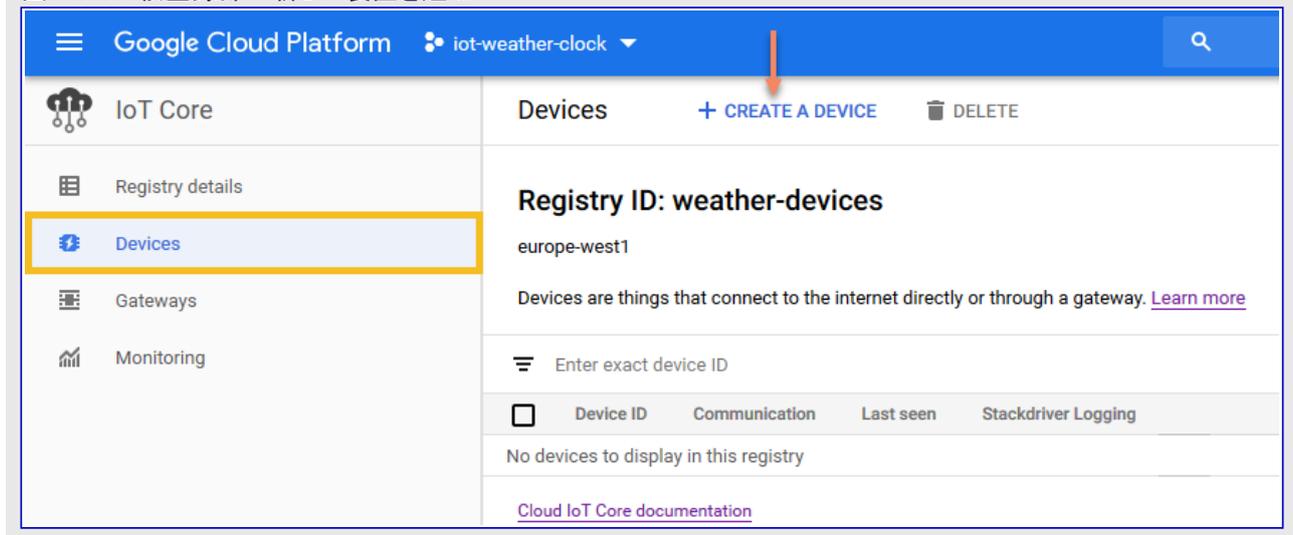
1.1.1. 装置を登録部に追加

登録部内の全ての装置は左手側のDevices(装置)タブを選ぶことによって見つかります。新しい装置はCreate Device(装置作成)を押すことによって追加することができます。図1-3をご覧ください。AVR IoT WG基板を追加するには”Device ID(装置ID)”、”Public key format(公開鍵形式)”、”Public key value(公開鍵値)”を除いて全てを既定のままにしてください。Device ID(装置ID)は”CLICK-ME.htm”ファイルのURLで見つかります。”CLICK-ME.htm”ファイルはキットがUSBを通して接続される時に”CURIOSITY”ドライブ下に置かれます。図1-2.で例をご覧ください。Googleクラウドは最初の文字が(数字ではなく)文字であることを必要とします。従って入力される装置IDは”d + 番号”であるべきです。例えば、”d0123710B94CEB0ECFE”です。

図1-2. IoTに対する装置ID発見

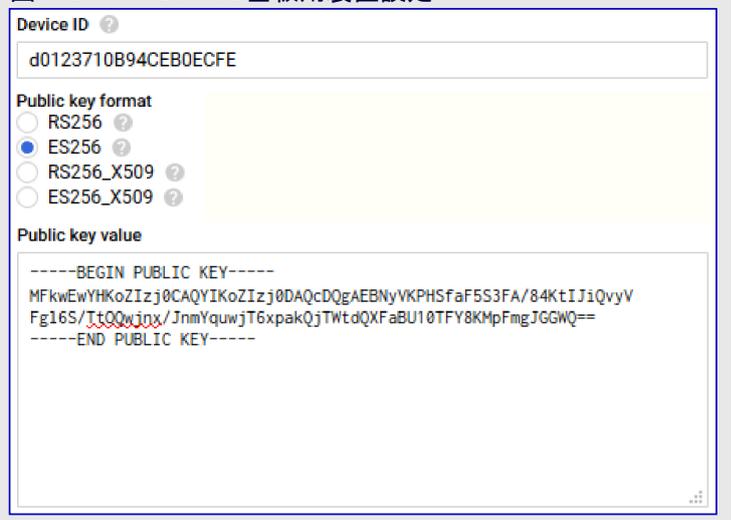


図1-3. IoT核登録部へ新しい装置を追加



公開鍵形式は”ES256”です。公開鍵はCURIOUSITYドライブ下の”PUBKEY.txt”ファイルで見つかります。内容をPublic key value (公開鍵値)領域に複写してください。詳細が図1-4.と同じであるべきです。装置を追加するために”Create(作成)”をクリックしてください。

図1-4. AVR® IoT WG基板用装置設定



1.2. クラウド関数設定

クラウド関数は起動(trigger)事象が起きる時に必ず走行するコード断片です。クラウド関数は音声命令要求を受け取り、それら进行处理し、そして処理したメッセージを装置に転送するのに使うことができます。新しいクラウド関数を作成するにはクラウド操作卓の左手側のメニューを通してそれを開くことによってCloud Functions(クラウド関数)単位部へ、続けて”Create Function(関数作成)”に誘導してください。クラウド関数設定での画面例については図1-5.をご覧ください。

表1-2.によって領域を埋めてください。

図1-5. クラウド関数作成

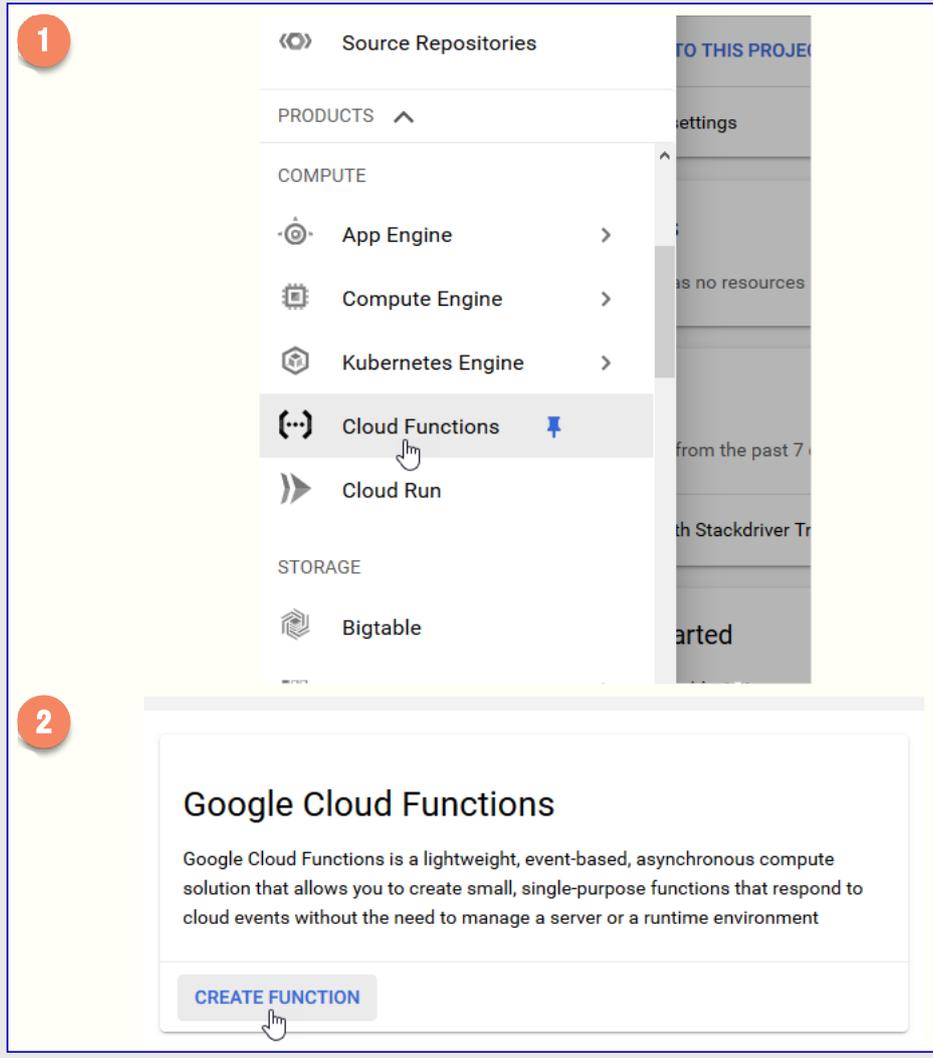
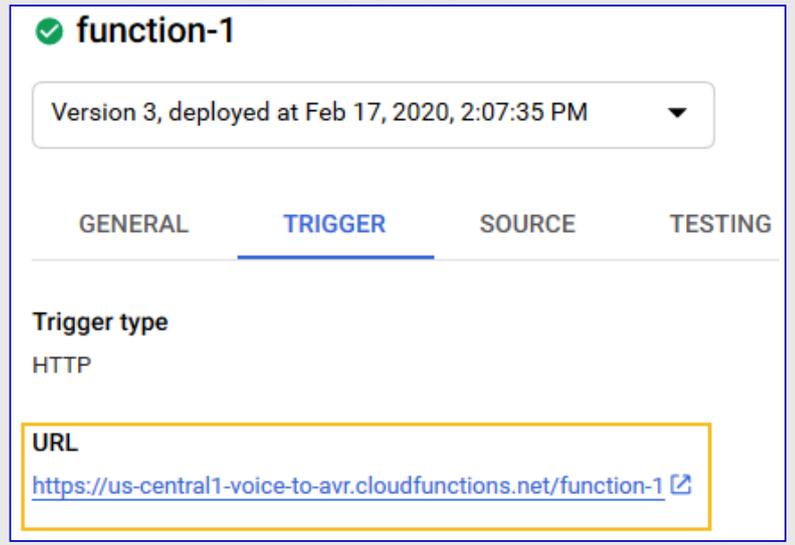


表1-2. クラウド関数領域

名前	入力	説明
Name	voice-command-process	クラウド関数を機別するための名前
Trigger	HTTP	クラウド関数の起動方法
Source Code	Inline editor	実行されるソースコード。"Inline editor"を使うことにより、ソースコードは下の領域で入力することができます。
Runtime	Python 3.7	ソースコードの言語
main.py	下のコードをご覧ください。	クラウド関数が起動される時に実行されるソースコード
requirements.txt	google-cloud-storage google-auth==1.6.2 google-api-python-client==1.7.8 google-auth-httpplib2==0.0.3 google-cloud-pubsub==0.39.1 paho-mqtt==1.4.0 pyjwt==1.7.1 oauth2client	ソースコードを走らせるのに必要とされるpython単位部
Function to excute	process_voice	ソースコードの入口点

クラウド関数が作成されたなら、それを開いてTrigger(起動)タブを選んでください。クラウド関数は呼び出しが一覧にされたURLに対して行われる時に必ず呼び出されます。画面例については図1-6をご覧ください。Dialogflowを構成設定する時に次の段階で使われるため、起動URLに注目してください。

図1-6. クラウド関数の起動URLを見つける場所



```

from googleapiclient import discovery
import base64

# 重要: これらの領域をあなたのプロジェクト設定に変更してください。
PROJECT_ID = "voice-to-avr"
IOT_CORE_REGION = "europe-west1"
IOT_CORE_REGISTRY_ID = "voice-devices"
IOT_CORE_DEVICE_ID = "d0123DFDAEF65AF85FE"

# https://cloud.google.comで得られたコード
def get_gcloud_client():
    api_version = 'v1'
    discovery_api = 'https://cloudiot.googleapis.com/$discovery/rest'
    service_name = 'cloudiotcore'

    discovery_url = '{}?version={}'.format(
        discovery_api, api_version)

    return discovery.build(
        service_name,
        api_version,
        discoveryServiceUrl=discovery_url,
        credentials=None,
        cache_discovery=False)

# https://cloud.google.comで得られたコード
def send_message_to_device(project_id, cloud_region, registry_id, device_id, payload):
    """
    Sends a message to an IoT Device through the config pubsub topic. (Config pubsub is /devices/
    d_id/config)
    :param project_id: Google Cloud project ID
    :param cloud_region: sWhich region is the device located in. For instance us-central1
    :param registry_id: IoT Core Registry the device is located in
    :param device_id: The device ID
    :param payload:
    :return:
    """
    client = get_gcloud_client()
    device_path = 'projects/{}/locations/{}/registries/{}/devices/{}'.format(
        project_id, cloud_region, registry_id, device_id)

    config_body = {

```

```

        'binaryData': base64.urlsafe_b64encode(
            payload.encode('utf-8')).decode('ascii')
    }

    return client.projects(
    ).locations().registries(
    ).devices().modifyCloudToDeviceConfig(
        name=device_path, body=config_body).execute()

def process_voice(request):
    request_json = request.get_json()
    queryResult = request_json['queryResult']
    parameters = queryResult['parameters']

    number = str(int(parameters['number']))
    payload = '{"number":{}}'.format(number)
    print("Sent {} to device".format(number))
    send_message_to_device(PROJECT_ID, IOT_CORE_REGION, IOT_CORE_REGISTRY_ID, IOT_CORE_DEVICE_ID,
    payload)

```

1.3. DialogflowとGoogle Actions

Dialogflowはどんなコード行も書く必要もなく、自然な言語処理とGoogleクラウド'応用へ音声制御の統合を許すサービスです。出力例について、Dialogflowは出力されるべき望む番号と共に「1.2. クラウド'関数設定」項で作成されたクラウド'関数用起動URLを呼び出すべきです。Dialogflowは<https://dialogflow.cloud.google.com>で見つけることができます。登録時、新しいAgent(取次)を作成する指示メッセージが現れます。それを”print-agent”と名付け、既定言語としてEnglish(英語)、Googleプロジェクトに前項で作成されたものを選んでください。取次を完成させるためCREATE(作成)をクリックしてください。作成処置の画面例については図1-7をご覧ください。

図1-7. Dialogflow取次作成

1.3.1. インテント作成

Dialogflowはインテント(Intent:意図、目的)の概念で動きます。インテントは使用者が行うことを望む何かで、意図する何かです。この例について、インテントはIoT基板からPCに与えられたメッセージを出力することです。インテントを作成するにはCreate Intent(インテント作成)をクリックしてください。最初の段階はいくつかの調教句(トレーニング フレーズ)を追加することで、これは使用者がそれらのインテントとどう話し合うかの自然言語での例です。以下の調教句(またはそれ以上)を追加してください。

- Please print the following 81727 1321 (以下を印刷(出力)してください。81727 1321)
- Could you print Twenty Four? (24を印刷(出力)してもらえますか?)
- Send 71 372 to the computer (71 372をコンピュータに送ってください。)

追加された全ての調教句について句の実体(entity)が記されなければなりません。実体はメッセージから抽出されるべき情報を定義する方法です。この例について、句内の番号を記して、句内の番号としてそれを識別する@sys.numberを選んでください。この手続きの画面例については図1-8をご覧ください。

図1-8. インテントに調教句追加

The screenshot is divided into two numbered sections:

1 Training phrases interface. A dropdown menu is open over the 'Add user expression' field, showing a list of system parameters: @sys.phone-number, @sys.music-genre, @sys.number-sequence, @sys.number (highlighted with a mouse cursor), @sys.zip-code, and @sys.given-name. A 'Filter' input is visible at the top of the dropdown. Below the list is a '+ Create new' button and a 'MANAGE PARAMETERS AND ACTION' link.

2 Intent resolution interface. It shows three training phrases with their extracted parameters:

- Phrase: "Send 71 372 to the computer".
- Phrase: "Could you print Twenty-Four?".
- Phrase: "Please print the following 81727 1321".

A table below the phrases shows the extracted parameters:

PARAMETER NAME	ENTITY	RESOLVED VALUE
number	@sys.number	71 372

Action(活動)とParameters(仮引数)の領域は抽出された実体がどの仮引数に割り当てられるべきかを定義するのに使われます。図1-9.で示されるように仮引数を埋めてください。

図1-9. 番号仮引数定義

The 'Action and parameters' interface includes an 'Enter action name' field and a table for defining parameters:

REQUIRED	PARAMETER NAME	ENTITY	VALUE	IS LIST	PROMPTS
<input checked="" type="checkbox"/>	number	@sys.number	\$number	<input type="checkbox"/>	Define prompts...
<input type="checkbox"/>	Enter name	Enter entity	Enter value	<input type="checkbox"/>	-

A '+ New parameter' button is located at the bottom left.

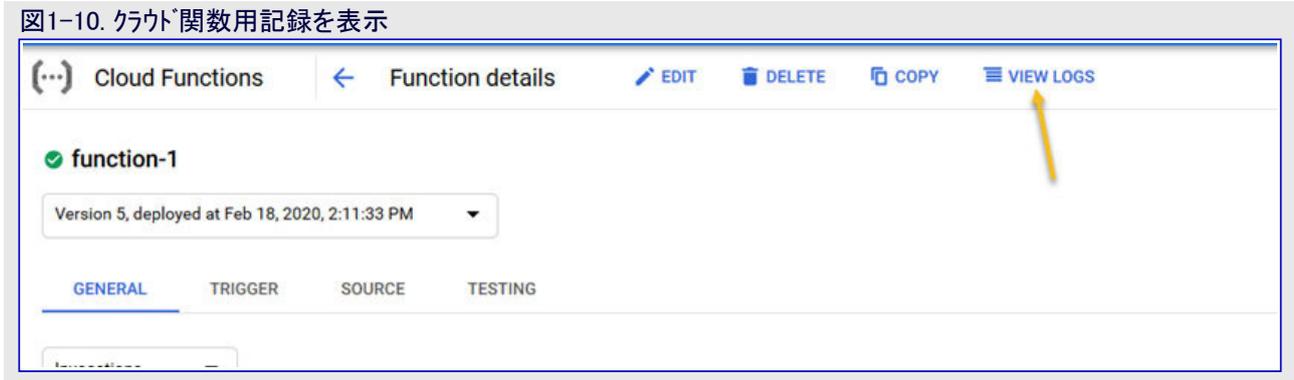
ダイアログにいくつかの変種を追加するため、応答を追加することができます。この応答はインテントが終了された時に使用者へ話返されます。Add Response(応答追加)をクリックし、"Printing \$number from the AVR-IoT Board(AVR-IoT基板から\$番号を出力(印刷))"と入力してください。"\$number"部分が上で定義された仮引数を参照することに注意してください。応答後に会話を閉じるにはSet this intent as end of conversation(このインテントを会話の最後として設定)をクリックしてください。最後に、Fulfillment(遂行)下でEnable webhook call for this intent(このインテント用Webhook呼び出しを許可)に対する摺動子を許可してください。右上隅でSave(保存)をクリックすることによって終了してください。

Fulfillment(遂行)は新しく得られた情報で会話の最後で対話が行うものです。左手側のメニューでFulfillment(遂行)をクリックしてWebhookを許可するために右手側の摺動子を使ってください。URL領域で以前の項で作成されたクラウド関数から起動URLを入力してください。他は既定のままとして"Save(保存)"をクリックしてください。

1.3.2. 対話の試験と確認

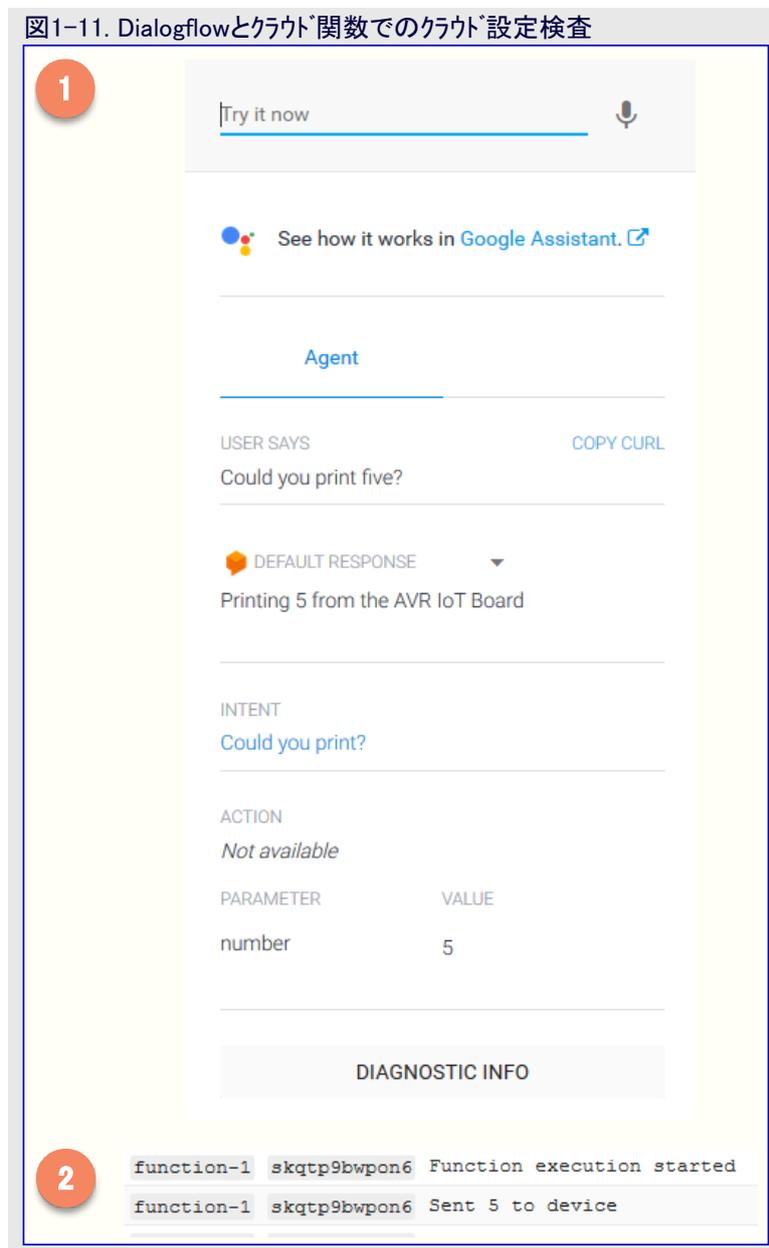
Homeアシスタント装置を追加してAVR-IoT基板を構成設定することによる手続きの前に、クラウドが正しく関数を設定したことを確認することが有益です。図1-10.で示されるように、「1.2. クラウド関数設定」項で作成したクラウド関数を開き、VIEW LOGS(記録を表示)をクリックしてください。

図1-10. クラウド関数用記録を表示



別のタブでDialogflowへ誘導してください。右上隅にTry it now(今すぐ試行)と名付けられた文章枠があります。ここで入力される文書は実際の音声命令を模倣します。例えば、Could you print five?(5を出力(印刷)してもらえますか?)と入力してください。これはクラウド関数に要求を送ります。図1-11.は全てが上手く行った場合の結果を示します。最初の画面例はDialogflowからで、2つ目はクラウド関数記録表示からです。

図1-11. Dialogflowとクラウド関数でのクラウド設定検査



1.3.3. Googleアシスタント対応装置の使い方

助言: 装置が”Hey Google”指示に応答するなら、きっとGoogleアシスタント対応が許され、この例に使うことができます。例えば、殆どのAndroid®フォンとGoogle Home装置は適合します。

Googleアシスタント対応装置は”Hey Google”指示で音声命令を発行するのに使うことができます。これを以前の項で作成したクラウド例で使うには同じGoogleアカウントで接続されなければなりません。Googleアカウントで接続する手順は当該装置に依存します。

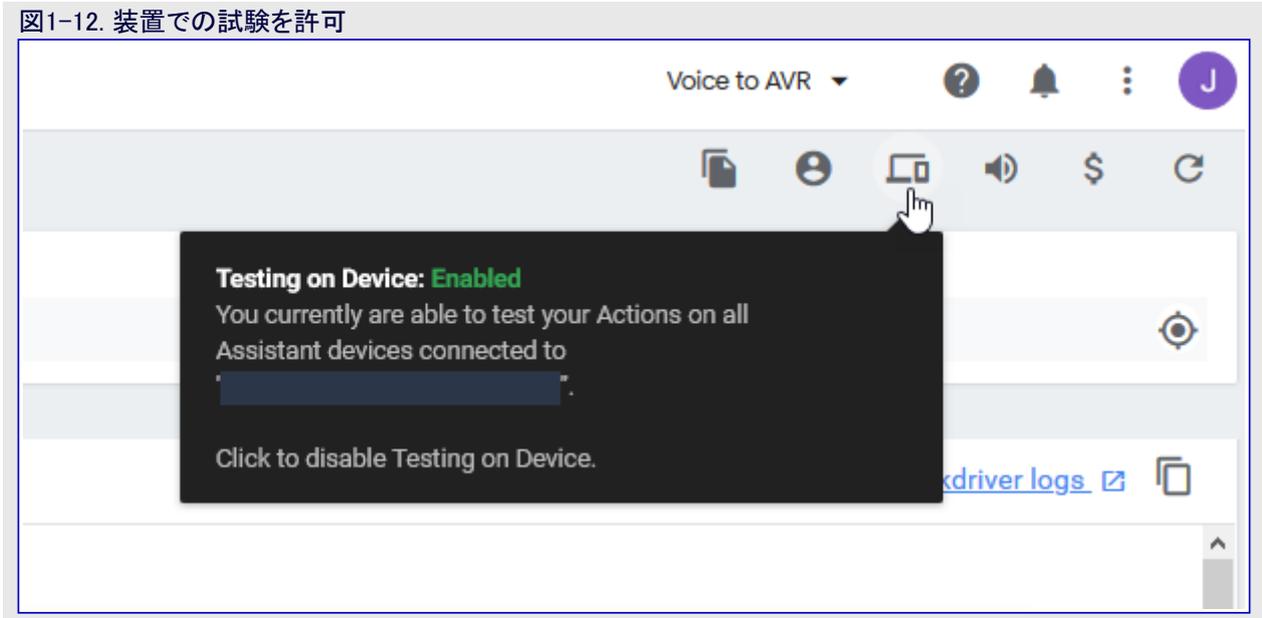
Googleアシスタントを使うには対話(Dialog)がDialogflowからGoogle Actionsにインポートされなければなりません。データはDialogflowのTry it now(今すぐ試行)文章枠直下の”See how it works in Google Assistant(Googleアシスタントの動作を確認)”リンクをクリックすることによってインポートされます。これは使用者をGoogle Actions操作卓へも向け直します。

活動(Action)はそれが名前を割り当てられる前に呼ぶことができません。名前は上部のバーで”Develop(開発)”をクリックして表示名を入力することによって割り当てられます。この例は”Microchip Voice”の名前を使います。変更を有効にするため、右上側のSave(保存)をクリックしてください。”Testing on Device(装置で試験)”設定は命令を実施するのに当該装置を使うことが許可されなければなりません。上部バーでTest(試験)タブへ誘導してください。図1-12.で示されるように、右上隅でTesting on Device(装置で試験)が許可されることを確実にしてください。

「1.3.2. 対話の試験と確認」で発行されたのと同じ音声命令が今や以下の対話によってアシスタント装置から発行することができます。Hey Google . . . Talk to Microchip Voice(Microchip Voiceに話し掛け) - これはMicrochip Voiceの試験版です。Greetings! How can I assist?(こんにちは!どのようなご用件でしょうか?) - Could you print five?(5を印刷(出力)してもらえますか?) - 5をAVR-IoT基板で出力します。

助言: 図1-10.で示されるように、クラウド関数記録を一瞥することによって正しい機能を確認することが常に可能です。

図1-12. 装置での試験を許可



2. AVR-IoT WG基板追加

助言: 読者が以降の本章で問題を持つ場合、処置する前にAVR-IoT WG基板使用者の手引きを読むことが推奨されます。これは<http://www.microchip.com/DS50002809>でダウンロードすることができます。必要な時に鍵となる概念の要約が与えられます。

助言: Atmel Studioは<https://www.microchip.com/mplab/avr-support/atmel-studio-7>で得ることができます。

出力(印刷)例の最終段階はクラウドからメッセージを受け取ってコンピュータに番号を出力するようにAVR-IoT基板を構成設定することです。AVR-IoT基板ファームウェアをインポートするにはAtmel Studioを開いてFile(ファイル)⇒New(新規)⇒Atmel Start Example Project(Atmel START例プロジェクト)を選んでください。”Board(基板)”引き落としメニューで”ATmega4809 AVR IoT WG”を見つけてください。選別された一覧で、図2-1.で示されるように、”AVR IoT WG Sensor Node(AVR IoT WG感知器節点)”、続いて”OPEN SELECTED EXAMPLE(選んだ例を開く)”をクリックしてください。

図2-1. Atmel StudioでIoT基板のファームウェアを発見



“Cloud Configuration(クラウド構成設定)”下で表2-1.のように詳細を入力してください。ソースコードをAtmel Studioにインポートするには“Generate Project(プロジェクト生成)”をクリックしてください。

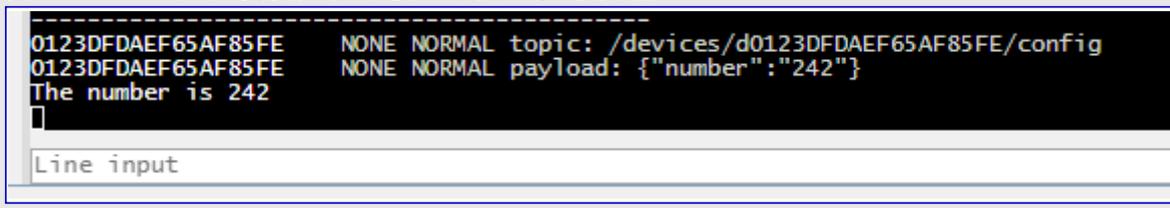
表2-1. IoT基板ファームウェア用クラウド構成設定入口

名前	入力	説明
Project ID	voice-to-avr	「1. クラウド構成設定」章で定義されたクラウドプロジェクトのID。このIDはProject Info(プロジェクト情報)下でGoogle操作卓の表紙でも見つけることができます。
Project Region	europa-west1	「1.1. IoT核構成設定」項で定義されたIoT核の地域。これはIoT核単位部表紙でも見つけることができます。
Registry ID	voice-devices	「1.1. IoT核構成設定」項で定義されたIoT核の登録部ID。これはIoT核単位部表紙でも見つけることができます。
MQTT Host	mqtt.googleapis.com	MQTTメッセージ用に接続するサーバー。無変更のままにしてください。

2.1. メッセージ処理

main.cではクラウドからの新しいメッセージが到着する時に必ずreceivedFromCloud関数が呼ばれます。「1. クラウド構成設定」で記述されたクラウド構成設定とで全ての音声命令メッセージはJSON文字列{"number": x}として送られます。このJSON文字列は解析されて番号xが抽出されなければなりません。抽出されたxはその後print呼び出しを通して即座にPCへ送られます。下のソースコードをご覧ください。MPLAB®データ可視器(Data Visualizer)のような道具を使うことにより、出力されたメッセージを見ることができます。構成設定した基板とで、図2-2.で示されるように、使用者が音声を通して出力(印刷)命令を発行する時に必ずデータ可視器に番号(数値)が現れます。

図2-2. MPLABデータ可視器で出力された結果の番号(数値)



```
// これは購読時にMQTTサーバーから配信されたメッセージを処理します。
void receivedFromCloud(uint8_t *topic, uint8_t *payload)
{
    debug_printer(SEVERITY_NONE, LEVEL_NORMAL, "topic: %s", topic);
    debug_printer(SEVERITY_NONE, LEVEL_NORMAL, "payload: %s", payload);

    char *numberToken = "¥number¥":";
    char *subString;

    if ((subString = strstr((char *)payload, numberToken)) {
        uint8_t numberStringLength = 6;
        char numberStr[numberStringLength];

        // 番号の開始位置
```

```
char *currentChar = &subString[strlen(numberToken)] + 1;

// ”に当たらない限りもっと桁があり、それらを記録
uint8_t i = 0;
while (*currentChar != ' ')
{
    numberStr[i] = *currentChar;
    currentChar++;
    i++;
    if (i > numberStrLength)
    {
        debug_printer(SEVERITY_WARNING, LEVEL_ERROR, "Number message has illegal
parameter: %s\n", subString);
        return;
    }
}
// 文字列を有効にするためヌル終端子を追加
numberStr[i] = ' 0';

// 番号出力(印刷)
printf("The number is %s\n", numberStr);
}
```

3. 改訂履歴

資料改訂	日付	注釈
A	2020年3月	初版文書公開

Microchipウェブ サイト

Microchipは<http://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<http://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<http://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、その製品系統が今日の市場でその種類の最も安全な系統の1つであると考えます。
- コード保護機能を破るのに使われる不正でおそらく違法な方法があります。当社の知る限りこれらの方法の全ては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です。おそらく、それを行う人は知的財産の窃盗に関与しています。
- Microchipはそれらのコードの完全性について心配されているお客様と共に働きたいと思います。
- Microchipや他のどの半導体製造業者もそれらのコードの安全を保証することはできません。コード保護は当社が製品を”破ることができない”として保証するとい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

デバイス応用などに関してこの刊行物に含まれ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。Microchipはその条件、品質、性能、商品性、目的適合性を含め、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**。Microchipはこの情報とそれの使用から生じる全責任を否認します。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、免責に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、Mmicrochipロゴ、Adaptec、AnyRate、AVR、AVRロゴ、AVR Freaks、BesTime、BitCloud、chipKIT、chipKITロゴ、CryptoMemory、CryptoRF、dsPIC、FlashFlex、flexPWR、HELDO、IGLOO、JukeBlox、KeeLoq、Kleer、LANCheck、LinkMD、maXStylus、maXTouch、MediaLB、megaAVR、Microsemi、Microsemiロゴ、MOST、MOSTロゴ、MPLAB、OptoLyzer、PacTime、PIC、picoPower、PICSTART、PIC32ロゴ、PolarFire、Prochip Designer、QTouch、SAM-BA、SenGenuity、SpyNIC、SST、SSTロゴ、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ロゴ、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ロゴ、memBrain、Mindi、MiWi、MPASM、MPF、MPLAB Certifiedロゴ、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。

Adaptecロゴ、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<http://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Googleアシスタントでの音声制御使用者の手引き(DS50002969A-2020年3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。

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: http://www.microchip.com/support ウェブアドレス: http://www.microchip.com	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 - 香港特别行政区 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: 45-4450-2828 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 - ガルピング 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 - デルネン 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 - マドリード 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
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 ホーストン 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			