



# AVR®住宅自動化キットでの開始に際して

## 序説

著者: Johan Lofstad, Microchip Technology Inc.

これはAVR® IoT住宅自動化キットに対する紹介の手引きです。キットをクラウト、に接続する方法、ファームウェアをダウンロート、して変更する方法、そして機能を追加する方法を記述します。キットが稼働すると、ステッピング、電動機駆動部が紹介されます。一般的なステッピング、電動機の原理、それらがどう動くかとそれをどう制御することができるかが網羅されます。最後の段階はキットに電動機をどう接続するか、それに対するトライバを書くのと電動機を遠隔的に制御するクラウト、上の独自制御盤の追加を網羅します。

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

序詞	<b>兑•••</b>		1
1.	開始(	こ際して ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	3
	1.1.	<b>ホストPCへの基板接続・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・</b>	3
	1.2.	AVR-IoTウェブ頁・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	3
	1.3.	Wi-Fi <sup>®</sup> 網への基板接続 ······	4
	1.4.	実時間でのクラウドデータ可視化 ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	5
	1.5.	他の設定の構成設定 ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	7
	1.6.	基板上書き込み器/デバッガインターフェース ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	8
	1.7.	Atmel STARTを使う実演の生成 ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	8
	1.8.	Atmel StudioへのAVR-IoT WG感知器節点STARTプロジェクトのエクスポート ・・・・・・・・・・・・・・・・・・・・・・・	1
2.	ファーム	<b>ウェア・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・</b>	1
	2.1.	実装:1桁送り返し・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	2
	2.2.	計画部 · · · · · · · · · · · · · · · · · · ·	3
3.	ステッビ	ンが電動機 : 基本 ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	4
	3.1.	ステッピング電動機の基本 ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	4
	3.2.	ステッピング電動機の制御 ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	5
4.	Stepp	er 2 Clickの使用 ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	5
	4.1.	Atmel STARTでPWMの追加 ······1	6
	4.2.	正しいPINMUXの選択 ····································	8
	4.3.	トライハを書く	8
_	4.4.	1 / フラウト に接続 ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	9
5.	改訂用	复 <b>歴</b> ••••••	2
Mic	rochip	<b>ካ፲ን</b>	3
製品	品変更	<b>通知サービス・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・</b>	3
おそ	客様支	援····································	3
Mic	rochip	デベイス コード保護機能 ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	3
法的	内通知	2	3
商村	票•••		4
品質	質管理	<u> </u>	4
世界	界的な	販売とサービス ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	5

# 1. 開始に際して

# 1.1. ホストPCへの基板接続

AVR-IoT WG開発基板は標準マイクロUSBケーブルを使ってコンピュータに接続することができます。一旦接続されると、基板の右上側角の LED配列が次の順で2度瞬間点灯すべきです。青色⇒緑色⇒黄色⇒赤色。基板がWi-Fi®に接続されないと、赤LEDが点灯します。 基板は下図で示されるように、ホストPC上で取り外し可能な記憶装置として現れるでしょう。それを開いて開始するためにCURIOSITY トライブをダブル クリックしてください。

注: 全ての手続きはWindows<sup>®</sup>、Mac OS<sup>®</sup>、Linux<sup>®</sup>環境で同じです。

図1-1.取り外し可能記憶	としてのCuriosity基板	
$\leftarrow$ $\rightarrow$ $\checkmark$ $\Uparrow$ 💻 $ ightarrow$ This PC		ບັ Search Th… ເ>
📌 Quick access	✓ Folders (7)	
ConeDrive	3D Objects	Desktop
CURIOSITY (D:)	Documents	Downloads
Network	Music	Pictures
	Videos	
	V Devices and drives (2)	
	Local Disk (C:) 38.5 GB free of 237 GB	CURIOSITY (D:) 1.01 MB free of 1.05 MB
	> Network locations (10)	

CURIOSITYドライブは以下の5つのファイルを含むべきです。

- ・CLICK-ME.HTM 使用者をAVR-IoTウェブ実演応用へ向け直し
- ・KIT-INFO.HTM 使用者を基板についての情報と資料を含むサ仆へ向け直し
- ・KIT-INFO.TXT PKOB nanoファームウェアと基板の通番についての詳細を持つ文字ファイル
- ・ PUBKEY.TXT データ暗号化に使われる公開鍵を含む文字ファイル
- ・STATUS.TXT 基板の状態状況を含む文字ファイル

Googleクラウト サント・ボックス アカウントをアクセスする専用ウェブ頁に行くためにCLICK-ME.HTMファイルをダブル クリックしてください。

## 1.2. AVR-IoTウェブ頁



図1-2.はAVR-IoT WGウェブ頁の画像を示します。この頁は感知器データを表示して使用者にWIFLCFGと呼ばれるWi-Fi証明書を再 生成することを許します。これは基板に設定することができ、アクセス ポイント パラメータを再構成設定するための記憶装置として働きま す。

下図で示されるように、頁中央の状態標識はシステム構成設定の進捗を示します。これらの標識は各段階が成功裏に完了される毎に1 度点灯します。



左端の標識は基板がホストPCに接続されたかを示します。その次のWi-Fi標識は一旦基板がWi-Fi網に接続されると点灯し、基板の 接続状態を示すために青LEDがONに切り替わります。Wi-Fi標識の右ではGoogleクラウト、メッセージキューイング、テレメトリトランスホート(MQT T:Message Queuing Telemetry Transport)標識が見つかり、Googleクラウト、ヘのTCPソケット接続とMQTT接続の状態を示します。最後に 右端の標識の点灯はデータが基板からサーバーへ流れていることを表します。これは成功裏の各データのMQTT発行に対して基板上の 黄色LEDの点滅によって示されます。

## 1.3. Wi-Fi<sup>®</sup>網への基板接続

### 1.3.1. AVR-IoTウェブ 頁経由

AVR-IoT開発基板をインターネットに接続するにはいくつかの方法があります。最も簡単な方法はAVR-IoTウェブ頁(www.avr-iot.com/)を 通すことです。このサイトの左下隅は使用者が(パスワート、不要の)網を開くための接続を選ぶ、または(WPA/WPA2/WEP)パスワート、保護さ れたWi-Fi®網に対する資格情報を入力する無線網接続ウィントウを示します。下図はこのウェブサイトでのWi-Fi資格情報の入力方法を 示します。

Ī

重要: Wi-Fi網のSSIDとパスワードは19文字に制限されます。空白で始まるまたは終わる引用符、名前、句の使用を避けてくだ さい。AVR-IoT開発基板は一連の2.4GHz網だけを支援し、従って、インターネットへの基板接続に可動性(モーバイル)ホットスポッ トを使うことが推奨されます。

ー旦必要とされる詳細が入力されたなら、Download Configuration(構成設定ダウンロート)釦をクリックしてください。これはホストPCにWIFI.CFG(文字)ファイルをダウンロート」します。基板のWi-Fi資格情報を更新するためにWIFI.CFGのダウンロート」位置からCURIOSITYトライブへそのファイルをドラッグ&ドロップしてください。一旦Wi-Fiアクセスポイントへの接続が成功裏に行われると、青LEDが点灯するでしょう。



重要: SSIDとパスワードの領域に入力されたどの情報もウェブ上または MicrochipやGoogleのサーハーに送信されません。代わりに、そ の情報はWIFI.CFGファイルを生成するために(ブラウザ内で)局 所的に使われます。

Wireless Network Log	gin
MCHP-IOT	
Your WiFi information is config file is generated in	not transmitted anywhere—the n your browser.
Network Type	
O Open	
WPA/WPA2	
○ WEP	
	Show passwor
	Download Configuration

#### 1.3.2. 命令行インターフェース(CLI)経由

Wi-Fi<sup>®</sup>に接続する別の方法はシリアル命令行インターフェース(CLI:Command Line Interface)を通すことです。このインターフェースは何れかのシ リアル端末応用を通してアクセスすることができます。「1.5.2. シリアルUSBインターフェース」で定義されるUART設定を使い、使用者はwifi命令を 入力することによって基板をWi-Fi網に構成設定することができます。図1-5.と図1-6.は各々、網を開くのと安全な網に接続するため の試行例を示します。wifi命令とそのパラメータのより多くの詳細については「1.5.2. シリアルUSBインターフェース」項を参照してください。

図1-6.シリアル命令行経由Wi-Fi®構成設定(安全な網)
COM7:9600baud - Tera Term VT
File Edit Setup Control Window Help
·
wifi MCHP.10T, microchip
▼

#### 1.3.3. ソフトAP経由

Wi-Fiに接続する最後の方法は基板上のWINC単位部の機能である高度なソフトウェア ア クセス ポイント(Soft AP:Software Access Point)動作を通すことです。この方法は使用者が ラップトップやPCの代わりに携帯電話やタブレットのような携帯装置だけを使う場合に理想 的です。ソフトAP動作は初期通電LED周回間に殆どの始動時間に対してSWO押し釦を 押し続けることによって移行することができます。ソフトAP動作が成功裏に移行されると、 基板をMCHP.IOT.ACCESSPOINTと名付けられたWi-Fiアクセス ポイントとして検出するこ とができます。ソフトAPが利用可能になると、青LEDが点滅を始めます。携帯電話やタブ レットのような携帯装置を使い、MCHP.IOT.ACCESSPOINTホットスポットに接続してくださ い。それは基板が接続する網のSSIDとパスワードを入力することができるサインイン頁に向 け直します。装置名は考慮されず、認証形式は常にWPA/WPA2 (2)です。一旦これら 詳細が入力されたなら、基板を網に接続するためにConnect(接続)釦をクリックしてくださ い。サインイン頁がどう見えるかを知るには右図を参照してください。

#### 図1-7. ソフトAP経由接続



### 1.4. 実時間でのクラウドデータ可視化

直ぐ使えるように全てのAVR-IoT開発基板はMicrochipのGoogleクラウト、サント、ボックス アカウントに予め登録されています。このアカウントは実演目的にだけ準備されています。AVR-IoT WG開発基板の感知器によって取り込まれた全てのデータはMicrochipサント、ボックス アカウントへ流され、右の詳細によって識別することができます。



Microchipサンドボックスアカウントに接続された基板によって流されたデータの恒久的な記録や収集はありません。データ記憶/保持のような Googleクラウト 機能の完全な記録は実演環境から一旦取り去られた基板を使って利用可能にでき、関連する装置ID/公開鍵は私的ア カウントに移行されます。

#### 1.4.1. Googleクラウト へのデータ発行

MQTT PUBLISHパケットは常に特定トピックを使ってMQTTブローカーへ送られます。AVR-IoT開発基板は通信で'/devices/{装置ID}/ev ents'トピックを使ってメッセージをGoogleクラウト'へ発行します。このトピックで発行されたメッセージは基板上の光と温度の感知器から得られ た実時間データを含みます。データはどの平均化も実行されず、これはウェブ頁で直ぐに見える変化を許すために行われます。PUBLIS Hパケットの送出頻度は使用者応用によって決めることができます。この応用は感知器データが1秒毎にクラウト、へ発行されるように書かれ ています。

#### 1.4.1.1. 発行されたメッセージの表示

ー旦基板がWi-Fiアクセス ポイントに接続されてクラウト、ヘのソケット接続が確立されると、AVR-IoTウェフ"頁は基板上の光と温度の感知器から獲得したデータの実時間図を示します。データは基板からクラウト、ヘJSONオブジェクトを通してMQTT PUBLISHハッケットとして送られます。 ASCII文字列は次のように形式化されます。

{'Light': XXX, 'Temp': YYY }、ここでXXXとYYYは10進表記で表現された数値です。基板上の黄色LEDは基板がデータを発行する ことを示すために1行毎に250ms間ONに切り替わります。



### 1.4.2. トピックの購読

データの発行に加えて、AVR<sup>®</sup>-IoT開発基板はトビックを購読する能力があります。トビックを購読すると、そのトビックを持つデータがフロー カーサーハーに発行される時に必ずGoogleクラウトからデータを受け取ります。接続された他のクライアント装置が購読されたトビックを使って データを発行することによってブローカーに送られる情報に受信者が興味ある時にトビックの購読が望まれます。SUBSCRIBEハ<sup>®</sup>ケット送出 後、購読の特定トビックで発行された全てのメッセージは基板によって受け取られます。今の時点で、基板は'/devices/{装置ID}/events' トビックを購読します。これはMQTT接続を使って購読するためにGoogleクラウトでよって提供される唯一のトビックです。

#### 1.4.3. メッセージの送出

www.avr-iot.comウェブ頁URLは光と温度の図の下に"Control your device(あなたの装置の制御)"部分を表示します。Toggle(交互切 り替え)釦はAVR-IoT基板へ切替値を送るのに使われます。

同様に、"Text(文)"部分は基板に文章文字列を送るのに使うことができます。ToggleとTextの領域は個別にまたは同時に変更することができます。これらの値はSend to device(装置へ送出)釦を押すことでだけ'/devices/{装置ID}/events'トピックを渡って発行されます。既定によって基板がこのトピックを購読するため、発行された全てのメッセージが基板によって受け取られます。

If your device is listening, you can use these controls to send it a base	e64 encoded JSON message, with any or all of these fields:
Toggle Sends values of 0 or 1 to device as "toggle" field	Text Sends string to device as "text" field microchip
Send to device	

## 1.4.4. 購読したトピックで受け取ったメッセージの表示

切替値交互切り替えはAVR-IoT開発基板上の黄色LEDに対する短く強制されるON/OFF状態に対応します。LEDは交互切替器の 位置に応じて短時間ON/OFFに留まり、その後にPUBLISHパケットを通して感知器データ送信を示すために点滅する通常の動きを再 開します。

Text(文)部分で入力されたメッセージは文字列の形式で基板に送信されます。黄色LEDの動きに加えて、Toggle(交互切り替え)とText (文)の領域の値が(Tera Term、Realterm、PuTTyなどのような)シリアル端末応用で見ることができます。

ž	図1-10. シリアル端末でのメッセージ表示	
	COM107:9600baud - Tera Term VT	x
	File Edit Setup Control Window Help	
	0123297DB2CDC154FE NONE NORMAL topic: /devices/d012300DB2CDC154FE NONE NORMAL payload: {"slider":50 "toggle":1,"text":"microchip"}	Â.

Microchipサンドボックスアカウントを通して接続した基板によって発行されたデータの恒久的な保管や収集はありません。Googleクラウトによって利用可能な完全な記憶機能は基板が実演環境(Microchipサンドボックス)を取り去られて私的アカウントに移行された後の使用者に対して利用可能です。

## 1.5. 他の設定の構成設定

AVR<sup>®</sup>-IoT開発基板が正しく直ぐに使えるように完全に書かれて準備されてやって来る一方で、使用者は未だUSBインターフェースを通して応用ファームウェアの動きの面を制御することができます。これを行うには次のように3つの方法があります。

- 1. 大容量記憶機能を使ってHexファイル(再書き込み)またはWIFI.CFG(資格再構成設定)のドラッグ&ドロップ
- 2. シリアル命令行インターフェース(CLI)を通す、またはMPLAB X IDEを使う命令
- 3. 基板上の書き込み器/デベッガPKOB nano

#### 1.5.1. 大容量記憶ドラッグ&ドロップ

組み込みデバイスを書く1つの方法はCURIOSITYドライブに.hexファイルをドラッグ&ドロップすることです。Cコンパイラッールチェーンは各プロシェク トに対する構築で.hexファイルを生成します。この.hexファイルはプロシェクトのコート、を含みます。Nano組み込みデバッガ(PKOB nano)はシリアル ポート インターフェース(シリアル-USBブリッジ)へのアクセスも提供します。これはファームウェア更新を含む変更された.hexファイルをドラッグ&ドロップする ことを容易にします。この機能はインストールされるどのUSBドライバも必要とせず、全ての主なOS環境で動きます。基板ファームウェア用の代 替応用、example.hex、ファイルはそれらが利用可能になり次第、AVR-IoTウェブ頁の下部のダウンロート、部分からの選択で利用可能です。

#### 1.5.2. シリアルUSBインターフェース

Wi-Fiアクセスポイント資格はAVR-IoT開発基板上のシリアル命令行インターフェース(CLI)を通して再構成設定することができます。このインターフェースは応用診断情報を提供するのにも使えるかもしれません。このインターフェースにアクセスするには何れかの好きなシリアル端末応用(即ち、Tera Term、CoolTerm、PuTTY)を使い、以下の設定でCuriosity Virtual COM port(Curiosity仮想COMポート)と呼ばれるシリアルポートを開いてください。

表1-2. シリアルUSBインターフェース設定								
ホーレート	データ	パリティ ビット	停止ビット	流れ制御	局所送り返し	送信規約		
9600	8Ľ vh	なし	1ビット	なし	ON	CR+LF (復帰+改行)		

注: Windows<sup>®</sup>環境の使用者について、USBシリアル インターフェースはMPLAB X IDEのインストールに含まれるUSBシリアル ホート トライバのインス トールを必要とします。

使用者は下表で一覧にされる命令キーワードを入力することによって基板を制御することができます。

衣 1-3. クリアル	ר וויערויל (דר ווייעריי) די אראי אין אין אין אין אין אין אין אין אין א				
命令	引数	説明			
reset	1	装置上の設定をリセット			
device	-	基板の固有装置IDを表示			
key	-	基板の公開鍵を表示			
reconnect	1	クラウドへの接続を再確立			
version	1	シリアル ポート使用者インターフェースのファームウェア版を表示			
cli_version	-	シリアル ポート使用者インターフェースの命令行インターフェース ファームウェア版を表示			
wifi	<網SSID>,<パスワード>,<保安任意選択*>	Wi-Fi網認証詳細を入力			
debug	<デバッグ任意選択 <b>**</b> >	基板動作の状態を見るためにデバッグメッセージを表示			

- \*: 認証形式任意選択は使われる網保安任意選択を決めるため以下の3つの番号の1つを入力することによって利用できます。
- 1. 開く パスワードと保安任意選択パラメータは必要とされません。
- 2. WPA/WPA2 保安任意選択は必要とされません。
- 3. WEP WEP網に繋ぐ時に網名、パスワード、保安任意選択(3)パラメータが必要とされます。例:'wifi MCHP.IOT,microchip,3'
- \*\*: デバッグ重要度任意選択は0~4の重要度番号を使うことによってdebug\_printer()に対して利用できます。
- 0.標準 SEVERITY\_NONEメッセージだけが表示されます。応用に於いて、これは購読トピック上で受け取った本体の表示です。
- 1. 警告 SEVERITY\_WARNINGと下のメッセージが表示されます。(注)
- 2. 通知 SEVERITY\_NOTICEと下のメッセージが表示されます。(注)
- 3. 情報 SEVERITY\_INFOと下のメッセージが表示されます。(注)
- 4. デバッグ SEVERITY\_DEBUGと下のメッセージが表示されます。このSEVERITYを使って複数のERROR処理支援メッセージが所定の場所にあります。
- 注: IoT WG開発基板応用ではこの形式のメッセージはありません。

#### 図1-11. シリアル命令行インターフェース

SCOM7:9600baud - Tera Term VT	
File Edit Setup Control Window Help	
help	
Unknown command. List of available commands:	
reset	
device	
key	
reconnect	
version	
cli_version	
wifi <ssid>[,<pass>,[authType]]</pass></ssid>	
debug	
	-

## 1.6. 基板上書き込み器/デバッガ インターフェース

MPLAB X IDEに精通した使用者については、これらのIDEの標準操作経由でAVR-IT基板を直接的に書き込みやデバッグを行うことができます。AVR-IoT開発基板はMPLAB Xによって自動的に検出され、基板上のPKOB nanoインターフェースを通して完全な書き込みとデバッグを許します。

## 1.7. Atmel STARTを使う実演の生成

革新的なオンライン ツールのAtmel STARTはAVR-IoT WG開発基板での独自実装に対して(これまでに入手可能な450の手本の中から) 100以上のClick感知器基板用単一Click支援を含む追加コート)例を選んで独自化するのに使うことができます。図1-12.で示されるよう に、コート、はAteml START頁でBROWSE EXAMPLES(例検索)をクリックすることによってダウンロート、することができます。Atmel STARTが AVR-IoT開発基板に対してコート、を生成するだけで、支援ライブラリはMPLABで未だ支援されず、PIC®デバイスは現在支援されていま せん。

AVR-IoT開発基板で使われるマイクロコントローラコードを生成するには、Atmel START先頭頁(https://start.atmel.com/)からBROWSE EXAMPLES(例検索)を選んで以下のこれらの簡単な手順に従ってください。

1. AVR-IoT WG Sensor Node(AVR-IoT WG感知器節点)開発基板を検索して選んでください(図1-13.)。

- 2. そのまま実演コードをダウンロードするには、DOWNLOAD SELECTED EXAMPLE(選択した例をダウンロード)をクリックしてください。コート を変更するにはOPEN SELECTED EXAMPLE(選択した例を開く)をクリックしてください(図1-13.、図1-14.)。
- 3. Googleクラウト プロジェクト詳細のような構成設定を変更するには、図1-15.で示されるように、頁をAVR IOT WG SENSOR NODE(AV R-IoT WG感知器節点)盤へ下スクロールしてください。使用者はhttps://console.cloud.google.com/cloud-resourcemanagerでプロジェクトIDのようなそれらのGoogleクラウト プロジェクト詳細を見つけることができます。

- 4. 一旦これらの変更が行われると、次の任意選択が利用できます。コート下見、後の使用のため構成設定保存、選んだ開発環境に プロジェクトをエクスポート。任意選択の1つを選ぶには、図1-16.で示されるように、頁の先頭の対応するタブをクリックしてください。
- 5. プロジェクトがエクスポートされるべきなら、EXPORT PROJECT(プロジェクトをエクスポート)タブをクリックし、使うIDEまたはツールを選んでください。その後にDOWNLOAD PACK(一括ダウンロート)卸をクリックしてください。一旦ダウンロートされたなら、Atmel STARTプロジェクトを Atmel Studioにインポートするのに「Atmel Studio 7での開始に際して」手引きに従ってください(図1-17.をご覧ください)。



#### 図1-13. Atmel START例検索

Atmel START Example proj information a Looking for ol	examples are a g ects are tailored bout each exam der examples? &	great starting point for embi for each compatible board a ple. Stmel Software Framework 3	edded programmers. and device, resulting contains previous re	Example projects will work "or in high quality production reac deases of non-configurable exa	ut of the box" but a dy code with a smal amples.	ire also easily to r Il memory footpri	nodify using Atmel STAI int. Read the available u	RT. Iser guides to get more
Search:	AVR JOT WO	<u>9</u>	Category:	All	~	Board:	All	
Name Categories		Description	Description			Board(s) supported User		
AVR IoT WG Sensor Node 🔶 🖓 💩			A sensor noo ATmega4808	A sensor node for the Google IoT Core Cloud based on the ATmega4808, WINC1510 and ECC608A.			VVR IOT WG	User guid
AVR IOT WG with AirQua	Sensor Node lity Click	<u> 70</u>	A sensor noo ATmega4808	de for the Google IoT Core Clou I, WINC1510 and ECC608A.	ud based on the	ATmega4808 /	WR IOT WG	User guid
AVR IoT WG Sensor Node 🔶 🕜 💩		A sensor nor ATmega4808	A sensor node for the Google IoT Core Cloud based on the ATmega4808, WINC1510 and ECC608A.			ATmega4808 AVR IoT WG		



#### 図1-15. AVR-IoT WG構成設定部分



## 図1-16. 使用者任意選択タブ



	A Restore To Found Rests 11 Mills And Pound
CODE     START ATTriega4808     SAVI     SA	CONFIGURATION
DOWNLOAD YOUR CONFIGURED PROJECT         Download a generated pack containing all your configured software components.         Select which IDE or command line tool you want the pack to include support files for:         Image: Select which IDE or command line tool you want the pack to include support files for:         Image: Select which IDE or command line tool you want the pack to include support files for:         Image: Select which IDE or command line tool you want the pack to include support files for:         Image: Select which IDE or command line tool you want the pack to include support files for:         Image: Select which IDE or command line tool you want the pack to include support files for:         Image: Select which IDE or command line tool you want the pack to include support files for:         Image: Select which IDE or command line tool you want the pack to include support files for:         Image: Select which IDE or command line tool you want the pack to include support files for:         Image: Select which IDE or command line tool you want the pack to include support files for:         Image: Select which IDE or command line tool you want the pack to include support files for:         Image: Select which IDE or command line tool you want the pack to include support files for:         Image: Select which IDE or command line tool you want the pack to include support files for:         Image: Select which IDE or command line tool you want the pack to include support files for:         Image: Select which IDE or command line tool you want tool you wa	WHAT TO DO NEXT? Use one of the selected IDEs and import your project as described in the user guide. If you do not have any IDEs installed you can download and install <u>Armel Studio 7.0 free</u> . Note: The exported pack can also be used later if you want to import a configuration Armel START Now to open in IDEs

## 1.8. Atmel StudioへのAVR-IoT WG感知器節点STARTプロジェクトのエクスポート

Atmel STARTでのAVR-IoT WGプロジェクト生成後、コンハイル、リンクされて最終的にAVRマイクロコントローラに書かれるためにそれをAtmel Studioヘエクスポートしてください。Atmel STARTプロジェクトをAtmel Studioヘインポートしてそれらを基板に書く方法の指示については「Atmel START使用者の手引き」を参照してください。

# 2. ファームウェア

Atmel STARTから実演をダウンロート、後、ファームウェアコートをアクセスすることができます。本章はファームウェアの概要ともっと重要な部分のいくつかの徹底的な調査を提示します。図2-1.は表2-1.での付随する記述と共に基本構造の大まかな構成図を示します。単位部がそれに対する点線を持つ場合、計画部は或る時点でその単位部から関数を呼びます。

表2-1. ファームウェア単位部	2-1. ファームウェア単位部					
単位部名	説明/目的	ソースで探す場所				
計画部	schedulerは最上部に属し、どの時間にどのルーチンを動かすかを制御	src¥timeout.c				
LED制御部	ファームウェアの内部状態を反映するようにLEDを設定/解除	led.c				
クラウト゛サーヒ゛ス(MQTT)	送受信両方に対してMQTTメッセージを処理	cloud¥*				
応用管理部	応用全体に対する最上位管理部。システムを初期化して計画部を開始	application_manager.c				
CLI	命令行インターフェース。USBを通してPCとの通信を処理	cli¥cli.c				
Wi-Fi階層	完全なWi-Fi階層。詳細な理解は必要とされません。	cloud¥wifi_service.c, winc¥*				
cryptoauthlib	完全な機能の暗号ライブラリ。詳細な理解は必要とされません。	cryptoauthlib¥*				
SPI	ATmega4809用の基本SPIドライバ	src¥spi_basic.c				
I2C	ATmaga4809田の簡単かI2Cドライバ	src¥i2c_master.c, src¥				
1-0		i2c_simple_master.c, i2c_types.c				
USARTとUSB	USB上にUSARTメッセージを書くためのドライハ	src¥usart_basic.c				



## 2.1. 実装:1桁送り返し

ファームウェアに独自応用を追加する最も簡単な方法はmain.cで見つかるsendToCloudとreceivedFromCloudの関数を通すことです。計 画部は或る時点でこれらの関数を呼び出してクラウドに対してデータを送って受け取ります。「1桁送り返し」使用事例はこの実演が紹介 されます。クラウドは装置に1桁を送ることができ、これは即座にクラウドへ返されて記録されます。

#### 手順1 : ファームウェア コード

メッセージがクラウトから受け取られると、receivedFromCloud関数が呼び出されます。受け取ったメッセージはJSON形式です。ファームウェアは この時点でJSON解析を支援していません。これは手動で行われなければなりません。この例では、番号を伴う通票のecho、例えば {"echo"."4"}を探すことによって解析されます。メッセージから番号が抽出されると、それは全域変数のecho\_numに割り当てられます。こ れは以下の方法で行われます。

```
uint8_t echo_num = 0;
void receivedFromCloud(uint8_t *topic, uint8_t *payload)
{
    char *echoToken = "¥"echo¥":";
    char *subString;
    if ((subString = strstr((char *)payload, echoToken))) {
        // "echo"に後続する1桁を抽出:、intへ変換するために'0'を減算
        echo_num = subString[strlen(echoToken) + 1] - '0';
    }
```

クラウドへの有効な接続がある限り、秒毎にsendToCloudが呼ばれます。既定ファームウェアは既に温度と光を記録します。JSONメッセージを 変更することにより、同時にecho\_num変数を送ることができます。下のコードはこれを実装し、秒毎にクラウドへ温度と光の値と一緒に更 新されたecho\_numを送り返します。

```
void sendToCloud(void)
{
    static char json[70];
    // この部分はCFG_SEND_INTERVAL秒毎に走ります。
    int rawTemperature = SENSORS_getTempValue();
    int light = SENSORS_getLightValue();
    int len = sprintf(json, "{¥"Light¥":%d, ¥"Temp¥":¥"%d.%02d¥", ¥"echo¥":%d}", light,
        rawTemperature / 100, abs(rawTemperature) % 100, echo_num);
    if (len > 0) {
        CLOUD_publishData((uint8_t *) json, len);
        LED_flashYellow();
    }
}
```

#### 手順2: クラウドから送出

AVR-IoTウェブ頁を開いて装置をインターネットに接続してください(「1.2. AVR-IoTウェブ頁」をご覧ください)。 echo番号は既に0の初期値を 持つ図として現れています。 Control your device(あなたの装置を制御)へ下スクロールし、新しい摺動子を追加してください。 摺動子を Min(最小)を0、Step(段階)を1、Max(最大)を9に設定してください。 これは2桁の番号が送られないことを保証します。 摺動子をechoに 改名してください。 正しい摺動子と出力については図2-2.をご覧ください。



## 2.2. 計画部

計画部(スケジューラ)はどの時間で何を動かすかを制御する単位部として前もって導入されています。各作業(タスク)は作業が走行する 前に経過しなければならなくて与えられた時間である経過時間をもちます。作業に対する時間が経過すると、計画部はCPUが利用 可能な時に必ずその作業を実行します。計画部の簡単化した状態図が図2-3.で示されます。





### 2.2.1. 作業(タスク)の作成と計画

いくつかの応用について、sendToCloudルーチンと別の間隔で走行する作業を持つことが望まれます。これは計画部に直接作業を追加することによって解決することができます。作業はscheduler\_timeout\_create関数呼び出しで計画されます。timer\_struct\_tへのポインタと制限時間の2つの引数が必要とされます。timer\_struct\_tは関数ポインタに加えて計画部に対する内部状態情報を保持します。計画部が作業走行準備可と判断すると、関数ポインタが呼ばれます。従って、この関数ポインタは作業へのポインタでなければなりません。

#### 手順1 : 作業関数の作成

目的は5秒毎に走行して基板上のLEDの1つを点滅する作業です。

```
static absolutetime_t led_task()
{
    LED_YELLOW_toggle_level();
    return 5000L;
```

'return 5000L;'に注目してください。これは作業が再び走行を計画されるのに先立つ刻時数です。

#### 手順2 : timer\_struct-t作成とそれの計画

```
int main(void)
{
    application_init();
    ENABLE_INTERRUPTS();
    timer_struct_t led_timer = {led_task, NULL};
    scheduler_timeout_create(&led_timer, 5000L);
    while (1) {
        runScheduler();
    }
return 0;
}
```

### 2.2.2. 考慮

計画部は定められた作業が最終期限前に走行するのを保証する優先権の概念を持ちません。換言すると、計画部は実時間計画部 ではなく、従って時間に敏感なシステムを動かすのに使うことはできません。例えば、4翼へリ(ドローン)の回転部制御です。4翼へりを浮か せる場合、回転部は同じ速度で動かなければなりません。回転部の速度を測定することにより、それらの速度が等しいことを確実に するため、実時間で異なる回転部の電力を調整することができます。これらの調整の各々間の時間が長すぎる場合、4翼へりは不安 定になって墜落します。従って、この制御繰り返しは実時間要件で、与えられた計画部で実装することはできません。

# 3. ステッピング電動機: 基本

以降の部分は簡単なステッピング電動機を理解するのに必要とされる基本的な原理とそれの制御方法を紹介します。ステッピング電動機の基本、それを回転する方法と速度のような回転特性を制御する方法が検討されます。そのようなパルスをどう生成するかの付随する検討と共にパルス制御例が提示されます。

### 3.1. ステッピング電動機の基本

ステッピング電動機は高いトルクと精度を提供するDC電動機の簡単な型です。ここで記述される電動機は最も簡単な型である双極永久 磁石ステッッピング電動機です。これは回転子としての永久磁石と固定子としてのいくつかの電磁石で動きます。異なる電磁石を励磁す ることにより、回転子は整列させられます。原理的な動作が図3-1.で図示され、完全な360°電動機回転を示します。

最初の段階は上下の電磁石を励磁することによって電動機を0°位置にします。電動機を90°に回転するため、励磁が左右に移動され、回転子をその位置に引っ張ります。この継続的な励磁交換が電動機を回転させます。

上で記述された動作は電動機が活性な電磁石の1対と完全に整列して移動するため、完全段階動作と名付けられます。1対よりもっ と励磁することにより、図3-2.で図解されるように、半段階動作を達成することが可能です。結果の力は2つの磁石の中央を示し、より 多くの正確な段階を許します。



各段階で電動機が移動する度数は段階角と呼ばれます。今まで検討された電動機は4つの磁石だけを持ち、半段階動作で45°の段 階角を許します。殆どの応用はもっと小さな段階角でもっとより滑らかな回転を必要とします。段階角はもっと磁石を追加することに よって段階角を小さくすることができ、これが殆どのステッピング電動機が4つよりももっと多く持つ理由です。



**助言**: ステッピング電動機のより多くの情報についてはhttps://www.microchip.com/wwwAppNotes/AppNotes.aspx?appnote=en0 12151をご覧ください。

## 3.2. ステッピング 電動機の制御

**重要**:本項は電動機を動かす方法を理解するのに必要ではありません。けれども、設計に於けるデバッグと拡張に対する「知って良かったこと」が考察されます。

ステッピング電動機の制御は正しい磁石を励磁することについ てが全てです。電磁石は電流則に従って励磁されます。巻 線によって誘導される起磁力はF=NIによって与えられ、ここ でIは巻線を通って行く電流、Nは巻線の巻き数です。要す るに、適切な巻線を通って流れる電流によって正しい磁石が 励磁されます。

図3-3.は1と2の2つの磁石対を持つ電動機に対する基本波 形を示します。各対の磁石はお互いに逆にされ磁力駆動を 作成します。2つ目の対は1つ目の対に比べて半期間ずらし ます。



力が固定の巻き数NでF=NIによって与えられるため、電流Iがトルクの量を制御する方法です。日よけを制御するなどのような多くの応用に対してマイクロコントローラは必要とされる電流を提供することができません。この問題を解決するのにHブリッジが使われます。

Hブリッジは磁石対毎に4つの切替器を設定することによってマイクロコントローラに波形の提供を許すと同時に電流供給を外部電源に許します。図3-4.はこの概念を図解します。磁石対上で正電流が必要とされると、切替器のS1とS4が閉じられます。各々S2とS3は負の 電流の流れを提供するために閉じることができます。



# 4. Stepper 2 Clickの使用

**重要**:本章はStepper 2 Clickドライハの実装を説明します。下で記述される例での完全なドライハ版はhttps://start.atmel.com/ #examplesのStepper 2 ClickでのAVR IoT WG感知器節点(AVR IoT WG Sensor Node with Stepper 2 Click)下でアクセス することができます。

本章はMikroElektronikaからのStepper 2 Click基板用ドライバを実装します。これは基本的なPWMインターフェースを提供するA4988ステッ ピング電動機制御器を利用します。表4-1.で提示される接続用の5つのピンがあります。

表4-1. Stepper 2 C	ē4−1. Stepper 2 Clickビン配列							
Mikroバスピン番号	ATmega4808ピン番号	略語	詳細					
1	PD7	CE	チップ許可 - チップが機能するにはLowでなければなりません。					
2	PA0	RST	リセット - このピンがLowなら、チップはリセットされます。					
3	PC3	SL	休止 - このピンがLowなら、デバイスは休止動作に移行します。					
16	PD4	ST	段階起動 - 各パルスに対して電動機を段階移動するPWMパルス					
15	PD6	DIR	方向 - ピンがHighなら、電動機は時計回りで回り、ピンがLowなら、電動機は反時計回りで回ります。					

## 4.1. Atmel STARTでPWMの追加



**助言**: ATmega4808でのPWM生成のより多くの情報についてはデータシート(https://www.microchip.com/wwwproducts/en/ATM EGA4809)の19章をご覧ください。

A4988はSTビンでPWM信号を受け取ることによって動き、電動機をハ<sup>ル</sup>ス毎に1段階移動します。AVR IoT WG基板回路図に従い、これはATmega4808のPD4ビンです。図4-1.で示されるように、デバイスのデータシートが述べるPD4は第4波形生成出力を通してPWMを支援します。

义	4-1	I. A	ſme	ga48	08データシ	ィートのピン暦	記列 − P	D4多重	化			
I	20	9	10	6	PD0		AIN0				0-WO0	
	21	10	11	7	PD1		AIN1	AINP3			0-WO1	
	22	11	12	8	PD2		AIN2	AINP0			0-WO2	
	23	12	13	9	PD3		AIN3	AINN0			0-WO3 📈	
	24	13	14	10	PD4		AIN4	AINP1			0-WO4	
	25	14	15	11	PD5		AIN5	AINN1			0-WO5	
	26	15	16	12	PD6		AIN6	AINP2				
	27	16	17	13	PD7	VREFA	AIN7	AINN2				
	28	17	18	14	AVDD							
	29	18	19	15	GND							
	30	19	-	-	PE0		AIN8			MOSI	0-WO0	

第4波形生成器出力はAtmel STARTで支援されるタイマ/カウンタA(TCA)を通して生成されます。Atmel Studioで前項から既存のIoT WGプロジェクトを開き、図4-2.で示されるようにAtmel STARTプロジェクトを再構成設定(Re-Configure)してください。

Solution 'AVR IoT WG Sensor Node With Stepper 2 Click' (1 project)   Build   Rebuild   Clean   Copy Full Path   Collapse   Scope to This   New Solution Explorer View   Add   Add   Add   Set as StartUp Project   Set as StartUp Project   Set as StartUp Project   View Project Snapshots   Take Snapshot   View Example Project Help   View Example Project Help   Set of the Export Project as Extension   Cut   Ctrl+X   Renove   Del   Rename   F2   Unload Project   Properties   Sensors handling.c	図4-2. Atmel	STARTプロジェクトの再構成設定			
Build   Rebuild   Clean   Clean   Copy Full Path   Collapse   Scope to This   Collapse   Scope to This   New Solution Explorer View   Add   Add   Add   Add Library   Add Arduino Library   Set as StartUp Project   Add Arduino Library   Corrigure Asinel Start Project   View Project Snapshots   Take Snapshot   View Example Project Help   Export Project as Extension   Cut   Ctrl+X   Renove   Del   Rename   F2   Unload Project   Properties   sensors_handling.c	💽 Solution 'A	WR IoT WG Sensor Node With Stepper 2	Click' (1 pro	ject)	≜
<ul> <li>Copy Full Path</li> <li>Collapse</li> <li>Scope to This</li> <li>New Solution Explorer View</li> <li>Add</li> <li>Add</li> <li>Add</li> <li>Set as StartUp Project</li> <li>Set as StartUp Project</li> <li>Add Arduino Library</li> <li>Set as StartUp Project</li> <li>Re-Configure Armel Start Project</li> <li>View Project Snapshots</li> <li>Take Snapshot</li> <li>View Example Project Help</li> <li>Export Project as Extension</li> <li>Export Project as Extension</li> <li>Cut</li> <li>Ctrl+X</li> <li>Rename</li> <li>F2</li> <li>Unload Project</li> <li>Sensors_handling.c</li> <li>sensors handling.h</li> </ul>		Build Rebuild Clean			
<ul> <li>Scope to This</li> <li>New Solution Explorer View</li> <li>Add</li> <li>Add Library</li> <li>Set as StartUp Project</li> <li>Set as StartUp Project</li> <li>Add Arduino Library</li> <li>Add Arduino Library</li> <li>Re-Configure Atmel Start Project</li> <li>View Project Snapshots</li> <li>Take Snapshot</li> <li>View Example Project Help</li> <li>Export Project as Extension</li> <li>Cut</li> <li>Ctrl+X</li> <li>Cut</li> <li>Ctrl+X</li> <li>K Remove</li> <li>Del</li> <li>Rename</li> <li>F2</li> <li>Unload Project</li> <li>Sensors_handling.c</li> <li>sensors handling.h</li> </ul>		Copy Full Path Collapse			
<ul> <li>Add</li> <li>Add Library</li> <li>Set as StartUp Project</li> <li>Set as StartUp Project</li> <li>Add Arduino Library</li> <li>Add Arduino Library</li> <li>Re-Configure Atmel Start Project</li> <li>View Project Snapshots</li> <li>Take Snapshot</li> <li>View Example Project Help</li> <li>Export Project as Extension</li> <li>Sensors Properties</li> <li>sensors handling.c</li> <li>sensors handling.h</li> </ul>		Scope to This New Solution Explorer View			
<ul> <li>Jet as startop Project</li> <li>Add Arduino Library</li> <li>Re-Configure A mel Start Project</li> <li>View Project Snapshots</li> <li>Take Snapshot</li> <li>View Example Project Help</li> <li>Export Project as Extension</li> <li>Cut</li> <li>Ctrl+X</li> <li>Remove</li> <li>Del</li> <li>Rename</li> <li>F2</li> <li>Unload Project</li> <li>Properties</li> <li>sensors_handling.c</li> </ul>		Add Library Set as Startl In Project			
<ul> <li>Ke-Configure A mer start Project</li> <li>View Project Snapshots</li> <li>Take Snapshot</li> <li>View Example Project Help</li> <li>Export Project as Extension</li> <li>Cut</li> <li>Ctrl+X</li> <li>Kemove</li> <li>Cut</li> <li>Rename</li> <li>F2</li> <li>Unload Project</li> <li>Properties</li> <li>sensors_handling.c</li> </ul>		Add Arduino Library			
Cut Ctrl+X Remove Del Rename F2 Unload Project Properties sensors_handling.c sensors handling.h	· •	View Project Snapshots Take Snapshot View Example Project Help Export Project as Extension	•		
Unload Project  Vroperties  Sensors_handling.c  Sensors_handling.h	© ⊾ •	Cut Remove Rename	Ctrl+X Del F2		
	c c b sen:	Unload Project Properties sors_handling.c sors_handling.h			•

"Add Software Component(ソフトウェア部品追加)⇒Timer(計時器)"をクリックすることによって計時器単位部を追加し、それを開いてください。正しい構成設定が図4-3.で示されます。本章の残り部分はこれらの設定が何かを記述します。

ATmega4808では計時器が分割動作に設定される場合、PD4でのPWMハ<sup>°</sup>ルス生成だけが可能です。計時器が分割動作の時は16ビット計時器を2つの8ビット計時器に効率的に分割し、その1つがWO4出力を生成します。分割動作で、上位ハ<sup>'</sup>イト定期(HPER)レジスタは PWM周期を定義し、上位ハ<sup>'</sup>イト比較1(HCMP1)レジスタはデューティサイクルを定義します。望む波形は一定周期を持ち、一定の速度で論 理値を反転します。例については図4-4.をご覧ください。これはHCMP1=HPER/2設定によって達成されます。約3msを与えるために 周期HPER=\$7Fが選ばれ、これの詳細は後述します。波形生成を許可するため、上位ハ<sup>'</sup>イト比較1出力値(HCMP1OV)を設定して上 位ハ<sup>'</sup>イト比較1許可(HCMP1EN)と許可(ENABLE)を許可するのに加えてPD4出力がWO4として選ばれます。

		MOTO TCA Init drive	R_TIMER r in Split Mode			
ENERAL	COMPONE	INT SETTINGS		СОМРО	NENT SIGNALS	
User guide	Driver:	Drivers:TCA:Init		~ wo/o:		
	Mode:	Split Mode		~ wo/1:		~
Rename component	CLOCKS			W0/2:		
Remove component	TCA:	Main Clock (CL)	(MAIN) (10 MHz)	W0/3:		~
					PD4	~
				WO/S-		
IVERS:TCA:INIT (SPLIT MODE) CONI	FIGURATION ON TC	A0	HIGH BYTE COMPAR	E CONFIGURATION	6	
ABLE: Module Enable: 0			HCMPOEN: High Con	npare 0 Enable: 0		
CSEL: Clock Selection: O Sys	stem Clock / 256	~	HCMP1EN: High Con	npare 1 Enable: 🧕	7	
GRUN: Debug Run: 0			HCMP2EN: High Con	npare 2 Enable: 🧧		
W BYTE COMPARE CONFIGURATION			HCMP0OV: High Cor Value:	npare 0 Output		
MPOEN: Low Compare 0 Enable: 🧕 🗌			HCMP1OV: High Con	npare 1 Output 🛛 😡	~	
MP1EN: Low Compare 1 Enable: 0			HCMP2OV: High Cor	nnare 2 Output		
MP2EN: Low Compare 2 Enable: 🛛 🗌			Value:			
MPOOV: Low Compare 0 Output 🛛 📃 ue:			HCMP0: Compare va	lue of channel 0: 🧿	0x0	he
MP1OV: Low Compare 1 Output 🛛 💽			HCMP1: Compare va	lue of channel 1: 0	0x3f	he
ue: MP2OV: Low Compare 2 Output			HCMP2: Compare va	lue of channel 2: 0	0x0	he
ue:			HCNT: High-byte Tin Register:	er Counter 🛛 😡	0×0	Pag
MP0: Compare value Channel 0: 0	0	Piex v	HPER: High-byte Per	iod Register: 🛛 😡	0x7f	he
MP1: Compare value Channel 1: 0 0x	0	hexy	INERRRUPT CONFIG	URATION		
MP2: Compare value Channel 2: 0 0x	0	hex v	Jockuda ISP baroarr	in driver, inc.		
NT: Low-byte Timer Counter 🛛 🛛 🕅	0	hes v	HUNF: High Underfle	w Interrupt		
ER: Low-byte Timer Period	ff	hexy	Enable:		_	
pister:			LCMP0: Low Compa Enable:	re 0 Interrupt 🛛 😡		
			LCMP1: Low Compar Enable:	re 1 interrupt 🛛 🛛		
			LCMP2: Low Compar Enable:	re 2 Interrupt 🛛 🛛		

図4-4. ステ	ッピング電動	」機Click用	目の望む	交互波制	3	

## 4.2. 正しいPINMUXの選択

各ピンに多重化された動作は左端側でAtmel STARTのPINMUX部分を選ぶことによって構成設定することができます。PD4ピンはTIMER(計時器)単位部によって既に構成設定され、構成設定されるべき表4-1.からの残りのピンを残しています。表4-2.に従って各ピンを構成設定してください。全てが構成設定された時に"Generate Project(プロジェクト生成)"を押してください。

#### 表4-2. Clicker 2用PINMUX構成設定

ピン名	通名	ピン動作	初期値
PD7	MOTOR_CHIP_EN	デジタル出力	Low
PA0	MOTOR_RST	デジタル出力	High
PC3	MOTRO_SL	デジタル出力	High
PD4	MOTOR_ST	デジタル出力	High
PD6	MOTOR_DIR	デジタル出力	High

#### 電動機の接続

Stepper 2 Clickは6つの入力、1A、1B、2A、2B、VIN、GNDを持ちます。これらは図4-5.の回路図に従って接続されます。マイクロコント ローラが充分な電流を供給できないため、外部電源が必要なことに注意してください。接続された基板の写真を図4-6.で見ることがで きます。





# 4.3. ドライバを書く

電動機はAtmel STARTを通して生成されたプロジェクトをフートで走行します。次の段階は電動機を開始/停止する関数を追加して速度 と方向を設定することです。電動機はタイマ/カウンタA(TCA)単位部を通して生成されたPWM信号の開始/一時停止によって開始/停止 します。電動機はTCA単位部の制御A(CTRLA)レジスタの許可(SPLIT\_ENABLE(訳注:データシートではENABLE))ビットを設定(1)すること によって開始されます。同じく、そのビットを解除(0)することによって停止されます。

```
void motor_start(void) {
    TCA0. SPLIT. CTRLA |= 1 << TCA_SPLIT_ENABLE_bp;
    // Stepper 2 Click許可
    MOTOR_CHIP_EN_set_level(false);
}
void motor_stop(void) {
    TCA0. SPLIT. CTRLA &= ~(1 << TCA_SPLIT_ENABLE_bp);
    // 計数器リセット
    TCA0. SPLIT. HCNT = 0;
    // Stepper 2 Click禁止
    MOTOR_CHIP_EN_set_level(true);
}</pre>
```

電動機の速度はPWMハ<sup>ル</sup>ルスの周期によって決められます。周期が増すと、速度が落ちます。使い易いインターフェースを持つため、速度 は最大と最小の周期の%として設定します。既定実装はTMAX=0.005s=5msの最大周期とTMIN=0.001s=1msの最小周期を持ちます。 目標周期(T)は入力した速度%に基づいてTMAXとTMINの間の値です。これは図4-7.で図解されるように、2点間の線上の点を見つけ 出すのと等価です。調整した周期は式4-1.の式で示されます。

```
式4-1. TMAXとTMIN間の周期T調整T = \frac{TMAX - TMIN}{100} \times 速度+TMIN
```



望む周期をHPERでの等価値に変換するには式4-2.で示されるように、クロック速度で周期を乗算してください。

# 式4-2. 周期に基づくHPER計算

HPER =  $T \times FS$ 

ここで、Fs=FCPU/256。用語Fsは周波数調整を表し、これは選んだクロック分周器なので、CPU周波数が256で割られます(「4.1. Atmel STARTでPWMの追加」をご覧ください)。先で検討されたように、均等なパルスを得るためにHCMP1=HPER/2です。

```
#define F_S (F_CPU/256)
#define T_MAX 0.005
#define T_MIN 0.001
#define DELTA_T (T_MAX - T_MIN)
void motor_set_speed(uint8_t speed) {
    // 速度(speed) = 100が最小周期を与えます。
    speed = 100 - speed;
    double period = (DELTA_T / 100.0) * (double) speed + T_MIN;
    uint8_t period_hper = F_S * period;
    TCA0. SPLIT. HPER = period_hper;
    TCA0. SPLIT. HCMP1 = period_hper / 2;
}
```

## 4.4. クラウト に接続

残りの全ては電動機をクラウドに接続することです。以下の部分を実装します。

- ・電動機の開始/停止の交互切り替え
- ・電動機の方向を設定する交互切り替え
- ・電動機速度を調整する速度摺動子

AVR IoTウェブサイで2つの交互切り替え、run(走行)とdirection(方向)を追加してください。0の最小値、1の段階の大きさ、100の最大 値を持つ1つのspeed(速度)摺動子を追加してください。これらの属性はJSON文字列、例えば、{"run":1,"direction":0,"speed":"74"} として装置へ送られます。正しい構成については図4-8.をご覧ください。装置はこの文字列を解析してそれによって動き、これは次の ように実装することができます。

```
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 *runToggle = "¥"run¥":";
    char *directionToggle = "¥"direction¥":";
    char *speedSlider = "¥"speed¥":";
    char *subString;

    // 電動機を開始/停止
    if ((subString = strstr((char *)payload, runToggle))) {
        if (subString[strlen(runToggle)] == '1') {
            motor_start();
        }else if(subString[strlen(runToggle)] == '0') {
            motor_stop();
        }else {
    }
}
```

```
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 *runToggle = "¥"run¥":";
   char *directionToggle = "¥"direction¥":";
   char *speedSlider = "¥"speed¥":";
   char *subString;
   // 電動機を開始/停止
   if ((subString = strstr((char *)payload, runToggle))) {
        if (subString[strlen(runToggle)] == '1') {
            motor_start();
        }else if(subString[strlen(runToggle)] == '0') {
            motor_stop();
        }else{
            debug_printer(SEVERITY_WARNING, LEVEL_ERROR,
                          "Supplied run command has illegal parameter: %s¥n", subString);
       }
    }
   // 電動機方向を設定
    if ((subString = strstr((char *)payload, directionToggle))) {
        if (subString[strlen(directionToggle)] == '1') {
            motor_set_direction(MOTOR_DIRECTION_CLOCKWISE);
        }else if(subString[strlen(directionToggle)] == '0') {
            motor_set_direction(MOTOR_DIRECTION_COUNTER_CLOCKWISE);
        }else{
            debug_printer(SEVERITY_WARNING, LEVEL_ERROR,
                         "Supplied direction command has illegal parameter: %s¥n", subString);
       }
   }
    // 速度設定
    if ((subString = strstr((char *)payload, speedSlider))) {
       uint8_t speedStrLength = 4;
       char speedStr[speedStrLength];
       // 速度番号の開始位置
        char *currentChar = &subString[strlen(speedSlider)] + 1;
       // "に当たらない限り、より多くの桁があります。それらを記録
       uint8_t i = 0;
        while(*currentChar != '"') {
            speedStr[i] = *currentChar;
            currentChar++;
            i++;
            if(i > speedStrLength) {
                debug_printer(SEVERITY_WARNING, LEVEL_ERROR,
                          "Speed command has illegal parameter: %s¥n", subString);
                return;
            }
        // 文字列を有効にするためにヌレ終了子を追加
        speedStr[i] = '¥0';
       // 速度文字列(speedStr)を整数に変換
        char *endptr;
        uint8_t speed = strtol(speedStr, &endptr, 0);
```

{

if (*endptr == '¥0') {     (/ 古字和合体の恋挽自由、声度乳字
// 又子列王评の发授成功、迷皮取足 http:/// 人子列王评の发授成功、迷皮取足 Noputing //2 to the second of the first second s
debug_printer(SEVERITY_NONE, LEVEL_NORMAL, "Setting speed %d \n", speed);
<pre>motor_set_speed(speed);</pre>
}else if(endptr == speedStr) {
// 変換中に何か不正、速度設定なし、警告発生
debug_printer(SEVERITY_WARNING, LEVEL_ERROR,
"The speed message is corrupted (conversion error): %s ¥n", speedStr);
}else{
// 文字列全体が変換され、何かが不正
debug_printer(SEVERITY_WARNING, LEVEL_ERROR,
"The speed message is corrupted (part conversion error): %s ¥n", speedStr);
debug printer(SEVERITY NONE, LEVEL NORMAL, "speed: %d", speed);
}

Toggles		Add Toggle
run 🕜		
direction 🕜		
Text Fields		Add Text Field
Sliders		Add Slider
speed 岱		Current: 74 🖞
Min 0	Step 1	Max 100

上限: Cでの文字列は一連の先頭文字へのポインタです。文字列は文字列の終了を示す0文字(¥0)に達するまでアドレスを増す ことによって続きます。より多くの情報はhttps://www.tutorialspoint.com/cprogramming/c\_strings.htmで見つけることが できます。

2つの交互切り替えは通票(例えば"run":)の開始を見つけることによって解析され、発生後の0または1のどちらかである単一文字を 抽出します。速度番号の抽出は未知の文字数(1、2、3文字)を含むためもっと扱い難いです。この問題は速度通票の最後の位置を 見つけることによって解決され、"文字に当たるまで全ての文字を読みます。"文字に当たると、全ての桁が抽出されてしまい、atoi() 関数を使って整数に変換することができます。

# 5. 改訂履歴

資料改訂	日付	注釈
А	2020年2月	初版資料公開

# Microchipウェフ゛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トウェア支援資料、最新ソフトウェア配布と保管されたソフトウェア
- ・全般的な技術支援 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/pen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 サイを通して利用できます。

# Microchipデバイスコート、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。
- コート、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トの安全を保証することはできません。コート、保護は当社が製品を"破ることができない"として保証すると言うことを意味しません。

コート、保護は常に進化しています。Microchipは当社製品のコート、保護機能を継続的に改善することを約束します。Microchipのコート、保護機能を破る試みはデジタルシ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、Mcicrochip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、MOSTロゴ、MPLAB、OptoLyzer、PackeTime、PI C、picoPower、PICSTART、PIC32ロゴ、PolarFire、Prochip Designer、QTouch、SAM-BA、SenGenuity、SpyNIC、SST、SSTロゴ、Super Flash、Symmetricom、SyncServer、Tachyon、TempTracker、TimeSource、tinyAVR、UNI/O、Vectron、XMEGAは米国と他の国に於 けるMicrochip Technology Incor poratedの登録商標です。

APT、ClockWorks、The Embedded Control Solutions Company、EtherSynch、FlashTec、Hyper Speed Control、HyperLight Load、 IntelliMOS、Libero、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、EC AN、EtherGREEN、In-Circuit Serial Programming、ICSP、INICnet、Inter-Chip Connectivity、JitterBlocker、KleerNet、KleerNet¤ ゴ、memBrain、Mindi、MiWi、MPASM、MPF、MPLAB Certified¤ゴ、MPLAB、MPLINK、MultiTRAK、NetDetach、Omniscient Code Generation、PICDEM、PICDEM.net、PICkit、PICtail、PowerSmart、PureSilicon、QMatrix、REALICE、Ripple Blocker、SAM-ICE、Se rial Quad I/O、SMART-I.S.、SQI、SuperSwitcher、SuperSwitcher II、Total Endurance、TSHARC、USBCheck、VariSense、View Sens e、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<sup>ロコ、</sup>、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AVR<sup>®</sup>住宅自動化キットでの開始に際して使用者の手引き(DS50002957A-2020年2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訳 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。



米国

# 世界的な販売とサービス

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 アトレス: http://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 インデアナポリス Noblesville, IN Tel: 317-773-8323 Fax: 317-773-5453 Tel: 317-536-2380 ロサンセ・ルス Mission Viejo, CA Tel: 949-462-9523 Fax: 949-462-9608 Tel: 951-273-7800 D-J- 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

オーストラリア - シト・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

亜細亜/太平洋

イント - ハンガロール Tel: 91-80-3090-4444 イント - ニューデリー Tel: 91-11-4160-8631 イント・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 ー ヘ・ナン Tel: 60-4-227-8870 フィリピン ー 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 ー バンコク Tel: 66-2-694-1351 ベトナム ー ホーチミン Tel: 84-28-5448-2100

亜細亜/太平洋

#### 欧州

オーストリア – ウェルス Tel: 43-7242-2244-39 Fax: 43-7242-2244-393 テンマーク - コヘンハーケン Tel: 45-4450-2828 Fax: 45-4485-2829 フィンラント – エスホー 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 トイツ – ガルヒング Tel: 49-8931-9700 ドイツ – ハーン Tel: 49-2129-3766400 トイツ - ハイルブロン 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 - ローセンハイム Tel: 49-8031-354-560 イスラエル - ラーナナ Tel: 972-9-744-7705 イタリア ー ミラノ Tel: 39-0331-742611 Fax: 39-0331-466781 イタリア ー パドバ Tel: 39-049-7625286 オランダーデルーネン Tel: 31-416-690399 Fax: 31-416-690340 ノルウェー - トロンハイム Tel: 47-72884388 ホ<sup>°</sup>ーラント<sup>゛</sup>ー 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 - マドリート Tel: 34-91-708-08-90 Fax: 34-91-708-08-91 スウェーデン – イェーテホリ Tel: 46-31-704-60-40 スウェーデン – ストックホルム Tel: 46-8-5090-4654 イキ・リス - ウォーキンガム Tel: 44-118-921-5800 Fax: 44-118-921-5820

Fax: 905-695-2078