

# AVR128DA48 Curiosity Nanoを使う差動ADC

## 事前要件

本項はこの訓練を完了するための全ての要件を提供する目的を持ちます。

### **ハート**ウェア事前要件

- AVR128DA48 Curiosity Nano (DM164151)
- Click boards™用Curiosity Nano Base (AC164162)
- ・POT Click基板 (MIKROE-3402)
- ・POT 2 Click基板 (MIKROE-3325)

注: POT Click基板は他の2つの可変抵抗器によって置き換えることができます。この場合の回路図が提供されます。

#### ソフトウェア事前要件

- ・MPLAB® X IDE 5.40版またはそれ以降
- ・ MPLAB XC8コンハ<sup>°</sup>イラ 2.20版またはそれ以降
- ・AVR-Dx\_DFP(デバイス系統一括) 1.1.40版
- ・MPLABデータ可視器(Data Visualizer)(MDV) 1.1版
- ・MPLABコート、構成部(MCC) 3.95版
- ・MCC 8ビットAVR MCUライブラリ 2.3.0版

#### 文書資料

- ・ AVR128DA28/32/48/64データシート
- TB3245:変換、累積、起動事象に関する12ビットADCの使用
- ・AVR128DA48デバイス概要で見つけることができる他のデバイス関連文書

## 序説

外部誘因の多くがアナログ型誘因であることを考慮すると、組み込み応用は度々アナログ感知器によって提供されるアナロ グ入力に依存します。殆どのMicrochipマイクロコントローラ(MCU)はアナログデータを採取するための統合されたA/D変換器 (ADC)が装備され、それを扱うことができます。

この文書はAVR128DA48 Curiosity Nano評価キットでADCの差動変換機能を使って応用を開発する方法を記述します。これは周辺機能の概要を提供し、差動動作でADCを構成設定するための手順を説明します。

この訓練完了後、使用者は以下をすることができます。

- ・全ての周辺機能を構成設定することによってシステムを初期化
- ADC変換を開始
- ・継続的にADC結果を読んで万能同期/非同期送受信器(USART)を通してそれを送出
- ・構成設定を理解するために応用を段階的に走行
- ハードウェア設定を使って開発した応用を試験
- ・図画的インターフェース道具を使ってデータを可視化

初めに、ADC単位部の概要が提示されます。その後、提起した応用が記述されます。概要確立後、応用を実装するのに必要とされる以下の全ての段階が提供されます。ソフトウェア道具の習熟、システムと周辺機能単位部の初期化、他の必要とされる機能の実装、応用のデバッグ、MPLABデータ可視器(Data Visualizer)を使って受け取ったデータの可視化。

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。

	次
	次

<b>其前要件 ••••••••••••••••••••••••••••••••••••</b>	1
	1
Гал 1	า บ
1.	კ ი
Ⅰ.Ⅰ. A/D変換奋車位部做安 1.0. c 田標画	კ ი
1.2. 心用概要	3
2. A/D変換器練習 ····································	4
2.1. アイコン識別子	4
2.2. ハート・ウェア準備 ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	4
2.3. ソフトウェア環境の習熟 ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	5
2.4. 課題 : 差動ADC ••••••••••••••••••••••••••••••••••••	8
3. 結び $\cdots$	8
4. 参考文献 · · · · · · · · · · · · · · · · · · ·	8
5 改訂履歴	8
	a
MCIOCINP / 1 / 1	0
	9 0
○各标文版 · · · · · · · · · · · · · · · · · · ·	9
/licrochipデバイス コ−ト 保護機能・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	9
去的通知 ····································	9
<b>奇標 ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・</b>	0
品質管理システム ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	0
世界的な販売とサービス ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	1

# 1. 概要

本章はADC単位部の概要とこの訓練の内容を提供します。

## 1.1. A/D変換器単位部概要

この文書で記述される応用を開発するのに使われたデバイスはAVR128DA48です。これにはシングルェントと差動の動作を提供する12 ビット分解能ADCが装備されています。

ADC入力信号はADCへの入力電圧が採取中に一定水準を保持されるのを保証する採取/保持回路を通して供給されます。ADC参照基準電圧はVREF周辺機能で構成設定されます。

ADC単位部の構成図が下で提示されます。



単位部を構成設定するには各々の単位部のレジスタが使われなければなりません。データシートの「レジスタ要約」と「レジスタ説明」の項が 単位部の全てのレジスタを一覧にして単位部レジスタの全てのビットとビット領域の機能を記述します。

この応用例ではADCが差動動作で構成設定されます。差動ADCは2つの入力間の差を測定します。これはいくつかの測定概念が 関心のある物理特性を数値化するため、1つの代わりに2つの出力信号を必要とするため、或る応用に於いて重要で有り得ます。こ のような概念を実装する感知器は一般的にそれらの出力値を差動信号としても知られる2つの信号間の電圧差として提供します。他 の感知器は例え測定それ自体がシングルエンド信号を生成しても、頑強性追加のために差動出力を提供するかもしれません。

差動アナログ感知器をMCUに接続する時に、差動対の信号の1つが正入力として定義され、同時に他方が負入力として定義されます。差動信号の値は負入力を参照基準とした正入力の電圧です。各信号の正と負の指示が差動信号の極性を決め、正入力が負入力よりも大きい場合に負として定義します。変換結果は次式によって与えられます。



ここでVAINPとVAINNは正と負のADC入力、VADCREFは選んだADC参照基準電圧です。差動変換のデータ形式は符号拡張付き2の補数です。

# 1.2. 応用概要

この応用を開発して試験するのに必要とされるハートウェア構成が図1-2.に描かれます。Curiosity Nano開発基板上のAVR128DA48デ バイスヘアナログ入力信号を提供するのに2つの可変抵抗器が使われます。ADCはこの入力信号間の電圧差を変換します。結果はUS ARTを通して送られ、MPLABデータ可視器(Data Visualizer)プラグインを使って作図されます。



# AVR<sup>®</sup> DA 訓練の手引き

この訓練用に設計されたソフトウェア応用は以下のソフトウェア構成図を実行します。システム初期化後、自由 走行するADC変換が開始されます。その後、無限繰り返しを使い、結果状態が継続的に読まれ、図画 的インターフェースを使って表示されるためにUSARTを通して送信されます。



# 2. A/D変換器練習

# 2.1. アイコン識別子

この副項はこの訓練とそれらの意味を通して使用者を手引きするのに使われるアイコンを提示します。右のアイコンが使われます。

1 情報:	このアイコンは有用な情報を強調するのに使われます。
(行うこと:	このアイコンは使用者が完了しなければならない作業(単位部 構成設定またはコード実装)があることを示すのに使われます。
★ 結果:	このアイコンは作業の解決策を記します。

# 2.2. ハート・ウェア準備

提供された訓練題材を開発するのに使われるハート・ウェアは図2-1.で提示されるAVR128DA48デハ・イス用Curiosity Nano開発基板から成ります。



POT Click基板とPOT 2 Click基板のように、この基板に対して有用な他の部品を容易に統合するのに、Curiosity Nano Base基板を 使うことができます。これは図2-2.で提示されます。



注: 試作実演を構築するのに簡単な可変抵抗器が使われます。可変抵抗器を容易に統合するため、Curisity Nano Base基板と共にMikroeのPOTとPOT 2のClick基板を使うことができます。

可変抵抗器は図2-3.で提示されるようにCuriosity Nano基板に接続されます。



## 2.3. ソフトウェア環境の習熟

本項は新しいプロジェクト作成に慣れることを助け、必要なプラグインを記述し、それをインストールする手順を提供します。

### 2.3.1. MPLAB<sup>®</sup> X IDEを使って新規プロジェクト作成

1. MPLAB X IDEを使って新しいプロジェクトを作成するには、File(ファイル)⇒New Project...(新規プロジェクト)へ行くとNew Project(新規プ ロジェクト)ウィサートが現れます。この段階は図2-4.で描かれます。

圛2	−4. 新規プロジェクト作成		
⊠	MPLAB X IDE v5.30 - avr128da48-cnano-adc	-example : pro	
File	Edit View Navigate Source Refactor Production D	ebug Team Tools	Window Help
2	New Project	Ctrl+Shift+N	🕨 þ • 🖳 • 🖳 • 🎧 🚯 • 🧰 🚥
2	New File	Ctrl+N	
2	Open Project	Ctrl+Shift+O	t Page 🗙 🛒 MPLAB X Store 🗴
	Open Recent Project		
	Import		
	Close Project (avr128da48-cnano-board-example)		
	Close Other Projects		DE
	Close All Projects		
	Open File		
	Open Recent File		
	Project Groups		S
	Project Properties (avr128da48-cnano-board-example)		ple
	Save	Ctrl+S	N
	Save As		acy Recent Projects
	Save All	Ctrl+Shift+S	built
	Page Setup		<pre><no pre="" recent=""  <=""></no></pre>
	Print	Ctrl+Alt+Shift+P	ล
	Print to HTML		
	Exit		J

2. プロジェクト選択段階に於いてCategories(区分)からMicrochip Embedded(Microchip組み込み)を、Project(プロジェクト)からStandalone Project(独立型プロジェクト)を選び、その後にNext(次へ)をクリックしてください。これは新しい独立型応用プロジェクトを作成します。この段階は図2-5.で描かれます。

図2-5. プロジェクト選択		
😵 New Project		×
Steps	Choose Project	
1. Choose Project 2	Q Filter:	
MPLAB	Categories: Microchip Embedded Other Embedded Samples	Projects: Standalone Project Existing MPLAB IDE v8 Project Prebuilt (Hex, Loadable Image) Project User Makefile Project Library Project Import START MPLAB Project Import Atmel Studio Project
XIDE	Creates a new standalone application your project.	project. It uses an IDE-generated makefile to build
	< Back	Next > Finish Cancel Help

3. 次の段階は書き込むデバイスを選ぶことです。この文書で提示される応用は8ビットAVR MCU系統からのAVR128DA48デバイスを 使って実装されます。このデバイス選択後、Next(次へ)をクリックしてください。

図2-6. デバイス選択		
😵 New Project		×
Steps	Select Device	
<ol> <li>Choose Project</li> <li>Select Device</li> <li>Select Header</li> <li>Select Tool (Optional)</li> <li>Select Plugin Board</li> <li>Select Plugin Board</li> <li>Select Project Name and Folder</li> </ol>	Family: 8-bit AVR MCUs (XMega/Mega/Tiny) ~ Device: AVR 128DA48 ~	
MPLAB XIDE		
	<pre></pre>	

4. その後、ツール選択段階でMicrochip Kits(Microchipキット)⇒AVR128DA48 Curiosity Nano (PKOB nano)から望むツール(SN:MCHP ~)を選んでください。これは任意選択の手順です。ツールが未だ利用できない、または未だ接続されていない場合、使用者はHar dware Tools(ハート・ウェア ツール)からSimulator(シミュレータ)を選ぶか、またはそれが必要とされるまで未選択のままにすることができます。

図2-7. ツール選択		
😰 New Project	>	<
Steps         1. Choose Project         2. Select Device         3	Select Tool (Optional)	
	< Back Next > Finish Cancel Help	

情報:使用者はそれに都合の良い名前(FN)を与えるためにツール名上でダブル クリックをすることもできます。

5. 図2-8.で提示されるように、このプロジェクトに対してXC8コンパイラ用の望む版が選ばれます。

図2-6.コアバイ )迭代 図2-6.コアバイ )迭代	×
Steps         1.       Choose Project         2.       Select Device         3.       Select Header         4.       Select Tool (Optional)         5.       Select Plugin Board         6.       Select Compiler         7.       Select Project Name and Folder	Select Compiler Compiler Toolchains AVR AVR (v5.4.0) [C:\Program Files (x86)\Atmel\Studio\7.0\toolchain\avr8\avr8-gnu-toolchain\b avrasm2 avrasm2 (v2.2.7) [C:\Program Files (x86)\Atmel\Studio\7.0\toolchain\avr8\avrassembler] XC8 \_XC8 (v2.10) [C:\Program Files (x86)\Microchip\xc8\v2.10\bin]
	< >>
	< Back Next > Finish Cancel Help

ĭ

6. 最後の段階はSelect Project Name and Folder(プロジェクト名とフォルダの選択)です。この例について、プロジェクト名はavr128da48-cn ano-adc-diffです。この段階中に使用者はProject Location(プロジェクトの場所)、Project Folder(プロジェクトのフォルダ)、Encoding(符 号化)を構成設定することもできます。

図2-9.7ロシェクト名と7オ 図 New Project	いずの選択		×
Steps	Select Project Name ar	nd Folder	
Choose Project     Select Device     Select Header     Select Tool (Optional)     Select Plugin Board     Select Compiler     Select Project Name and     Folder	Project Name: Project Location: Project Folder:	avr128da48-cnano-adc-diff       C:\Users\AVRist\MPLABXProjects       MPLABXProjects\avr128da48-cnano-adc-diff.X	
MPLAB X IDE	Overwrite existin Also delete sourd Set as main proj Use project locat Encoding: ISO-	ng project. ces. ject tion as the project folder 8859-1 V	
		< Back Next > Finish Cancel He	lp

必要とされる全ての段階を完了後、新しい空のプロジェクトがProject(プロジェクト)ウィントウで現れます。必要な全てのファイルはMCCプラクインを使ってプロジェクトに導入されます。

### 2.3.2. MPLAB<sup>®</sup>⊐ート<sup>\*</sup>構成部(MCC)

このプロジェクトを開発するのにMCCが使われます。これはプロジェクトに挿入されるべき理解し易いCコードを生成する図画的プログラム作成環境です。これは直感的なインターフェースを提供し、豊富な周辺機能一式と望む応用に特化した関数を許可して構成設定します。

このプラグインは「MPLAB®コード構成部(MCC)のインストール」ウェブ頁で提供される手順に従ってインストールすることができます。

必要とされる全てのソースとヘッダのファイルの生成後、使用者は望む機能を得るためにソースコードを完了しなければなりません。コート・実装後、応用の機能は試験されなければなりません。これはMDVプラグインを使って達成されます。

### 2.3.3. MPLAB<sup>®</sup>データ可視器 (MDV)

MDVはMPLABプラグインまたは独立型デバック ツールとして利用可能な図画的走行時デバック ツールです。これは組み込み応用内の変数と関数を走行時に図画的に表示します。

データは次のように図表化することができます。

- ・単一8ビット変数生流し
- ・データ流し規約での複数変数

MDVプラグインをインストールするにはMCCインストール手順を使うことができ、プラグイン名だけが違いです。このツールは応用実装後にADC結果を表示するのに使われます。

### 2.4. 課題 : 差動ADC

#### 2.4.1. 単位部初期化

この副項はADCを差動動作で使い、USARTを通して結果を送信し、図画的インターフェースでデータを分析するのに使う簡単な応用を開発するのに必要な手順を提供します。

望む全ての単位部を初期化するのにMCCが使われます。プラグインを開くにはWindow(ウィントウ)→MPLAB Code Configurator(MPLA Bコート<sup>\*</sup>構成部)→MPLAB Code Configurator Open/Close(MPLABコート<sup>\*</sup>構成部を開く/閉じる)へ行くか、または図2-10.で提示される MCCアイコンをクリックしてください。

図2-10. MCCを開く/閉じる	
Team Tools Window Help	
• 🕨 - 🔁 - 🎦 - 🏹 🌇 - 🤓	PC: 0x0 ithsvnzc
Kit Window × Start Page ×	MPLAB® Code Configurator v3: Open/Close

MCCを使ってこの応用を開発する最初の段階は初期化設定を提供してプロジェ外に望まれる周辺機能単位部を追加することです。 使用者は望む構成設定で必要な全ての周辺機能を初期化することによってシステムを初期化しなければなりません。その後、生成さ れたソースとヘッダのファイルを使い、使用者はMCUによって実行される算法を実装しなければなりません。

### 2.4.1.1. デバイス システム構成設定

ADC応用を開発する最初の段階はシステムクロックを構成設定することです。どれがクロック任意選択かを見つけ出すため、使用者はデバ イスのデータシートで「CLKCTRL - クロック制御器」章を調べなければなりません。

例えば、この応用では選ばれた主クロック周波数が2MHzです。

行うこと: MCCを使ってクロック元として内部高周波数発振器を構成設定してください。この発振器の周波数は4MHzでなけれ ばなりません。主クロック周波数は2MHzでなければなりません。

結果: MCCを開いてResource Management [MCC](資源管理)⇒Project Rersources(プロジェクト資源)⇒System(システム)⇒Syste m Module(システム単位部)へ行ってください。Easy Setup(容易な設定)タフ で以下の構成設定が行われなければなりませ ん。

- ・ Clock Control(クロック制御)
  - Clock Source(クロック元)
  - Internal Oscillator Frequency(内部発振器周波数)
- Prescaler Enable(前置分周器許可)
- Prescaler(前置分周器)

この設定は図2-11.で描かれます。

: 1-32 MHz internal oscillator(1~32MHz内部発振器)

: Internal Oscillator(クロック制御)

- Oscillator Frequency Options(発振器周波数任意選択): 4 MHz system Clock (default)(4MHzシステム クロック(既定))
  - : チェック
  - :2X(2分周)

### 図2-11. MCCクロック制御構成設定

Clock Control		
Main Clock(Hz):	2000000	
Clock Source :	Internal Oscillator	•
Internal Oscillator Frequency:	1-32MHz internal oscillator	•
Oscillator Frequency Options:	4 MHz system clock (default)	-
PLL Enable:		
External Clock(Hz):	1 ≤ 1000000 ≤ 20000000	
Prescaler Enable:	$\checkmark$	
Prescaler:	2X	-
Clock Out Enable:		

### 2.4.1.2. ADC構成設定

この文書で提示された応用について、ADCは差動動作で構成設定されます。MCCを使ってこの単位部の初期化設定を構成設定す るため、使用者はProject Resources(プロジェクト資源)にこの単位部を追加しなければなりません。

行うこと: MCCでADC周辺機能単位部をProject Resources(プロジェク外資源)に追加してください。

結果: Resource Management [MCC](資源管理[MCC])⇒Device Resources(デハイス資源)⇒Peripherals(周辺機能)⇒ADCへ行 き、緑の■符号をクリックすることによってADC周辺機能単位部を追加してください。この単位部がProject Resources(プロ ジェクト資源)ウィントウに現れます。

次の段階はADC単位部を構成設定することです。ADCは差動動作で走行し、採取した試料を継続的に変換します(自由走行動 作)。分解能は12ビット、前置分周器値は4、差動入力はPD3(正入力)とPD4(負入力)です。



- ・差動動作変換、自由走行を許可
- ・結果分解能を12ビットに設定
- ・500kHzのADC周波数を得るようにADC前置分周器を構成設定
- ・ADC入力ピン構成設定: 負入力としてADC入力4ピン、正入力としてADC入力3ピン

結果: Resource Management [MCC](資源管理)⇒Project Rersources(プロジェクト資源)⇒Peripherals(周辺機能)へ行き、ADC0 を選んでください。Easy Setup(容易な設定)タブで以下の構成設定が行われなければなりません。

- ・ Software Settings(ソフトウェア設定)
  - Result Selection(結果選択)
- : 12-bit mode(12ビット動作)
- Differential Mode Conversion(差動動作変換): enabled(許可)
- Left Adjust Result(結果左揃え)
- : チェックなし
- Hardware Settings(ハートウェア設定)
  Enable ADC(ADC許可)

この設定は図2-12.と図2-13.で描かれます。

: チェック

<ul> <li>Software Settings</li> </ul>	
API Prefix:	ADC0
Result Selection :	12-bit mode
Oifferential Mode Conversion :	enabled -
Optimized in the second sec	
図2−13. ADC0ハート・ウェア設定 ▼ Hardware Settings	
図2-13. ADC0ハート・ウェア設定 ▼ Hardware Settings	✓

500000

0 ≤ 0

No accumulation

Ŧ

≤ 31

加えて、使用者はADCレシ、スタを構成設定しなければなりません。どのレシ、スタで何の設定が利用可能かを知るのとレシ、スタのビットとビット 領域の利用可能な構成設定を見つけるため、使用者はデバイスのデータシートを調べなければなりません。

ADC Clock(Hz):

③ Sample Accumulation Number:

Sample Length (# of ADC Clock) :

自由走行任意選択はADC単位部の制御A(CTRLA)レジスタで利用可能です。CTRLAレジスタを書くことにより、使用者はスタンハイ動作で走行を許可/禁止、変換動作を構成設定、結果整合の設定、分解能の選択、周辺機能の許可/禁止を行うことができます。この周辺機能のクロック周波数は制御C(CTRLC)レジスタの前置分周器(PRESC)ビット領域を使って構成設定されなければなりません。

ADCを差動動作で使うには2つのアナログ入力が必要とされます。正入力は正ADC入力用多重器選択(MUXPOS)レジスタを使って構成 設定されなければならず、同様に負入力は負ADC入力用多重器選択(MUXNEG)レジスタを使って構成設定されなければなりません。 各々のピンは禁止されたデジタル緩衝部を持つアナログ入力としても構成設定されなければなりません。

レジズタの構成設定はMCCを使ってRegisters(レジズタ)タブで行うことができます。

• Register: CTRLA

- FREERUN(自由走行): enabled(許可)

図2-14. ADC0 C	「RLAレシ <sup>゛</sup> スタ	
💂 Register: CTRLA	0x23	
		· · · · · · · · · · · · · · · · · · ·
CONVMODE	enabled	•
	blad	
W ENABLE	enabled	· ·
FREERUN	enabled	<b>*</b>
	chabica	
LEFTADJ	disabled	<b>•</b>
RESSEL	12-bit mode	·
RUNSTBY	disabled	-

・Register: CTRLC	図2-15. ADC0 CTRLCレジネタ
- PRESC(前置分周器): CLK_PER divided by 4	▼ Register: CTRLC Ox1
(4分周CLK_PER)	● PRESC CLK_PER divided by 4 ▼
<ul> <li>Register: MUXNEG</li> <li>MUXNEG(多重器負選択): ADC input pin 4 (ADC入力4ヒ<sup>°</sup>ン)</li> <li>Register: MUXPOS</li> <li>MUXPOS(多重器正選択): ADC input pin 3 (ADC入力3ヒ<sup>°</sup>ン)</li> </ul>	図2-16. ADC0 MUXNEGとMUXPOSのレジネタ Register: MUXNEG 0x4  MUXNEG ADC input pin 4 ・  Register: MUXPOS 0x3  MUXPOS ADC input pin 3 ・

### 2.4.1.3. VREF構成設定

ADC参照基準電圧はADC0参照基準(ADC0REF)レジスタを書くことによって構成設定することができます。これはMCCを使って構成設定することができます。



**行うこと**: MCCを使ってVREF単位部をProject Resources(プロジェクト資源)に追加してADC参照基準電圧を2.048Vに構成設定 してください。

結果: Resource Management [MCC](資源管理[MCC])⇒Device Resources(デバイス資源)⇒Peripherals(周辺機能)⇒VREFへ 行き、緑の■符号をクリックすることによってVREF周辺機能単位部を追加してください。この単位部がProject Resources (プロジェクト資源)ウィントウに現れます。

Resource Management [MCC](資源管理)⇒Project Rersources(プロジェクト資源)⇒Peripherals(周辺機能)へ行き、VREF を選んでください。Easy Setup(容易な設定)タブで以下の構成設定が行われなければなりません。

・ Hardware Settings(ハートウェア設定)

- ADC Voltage Reference(ADC参照基準電圧): Internal 2.048V reference(内部2.048V参照基準)

図2-17. VREF ADC参照基準電圧

ADC Voltage Reference:
 Internal 2.048V reference

#### 2.4.1.4. USART構成設定

ADCを使って受け取ったアナログデータを変換した後、更にデータは使用者によって分析されるため送ることができます。データを開発しているホストコンピュータへ送るのにUSART周辺機能単位部が使われます。AVR128DA48 Curiosity Nano基板上でUSART1のRX(受信)とTX(送信)のピンは直接デバッガのピンに接続され、故に使用者は追加の線なしにデータをコンピュータへ送ることができます。従って、この応用で使われるUSART周辺機能単位部はUSART1です。

Y 行うこと: MCCを使ってプロジェクトにUSART1単位部を追加してください。

結果: Resource Management [MCC](資源管理[MCC])⇒Device Resources(デバイス資源)⇒Peripherals(周辺機能)⇒USARTへ 行き、緑の■符号をクリックすることによってUSART1周辺機能単位部を追加してください。この単位部はProject Resource s(プロジェクト資源)ウィントウに現れます。

プロジェクトに単位部導入後、いくつかの初期構成設定が行われなければなりません。

**行うこと**: USART1を9600のボーレート、パリティなし、1停止ビットの非同期動作での走行に構成設定してください。文字の大きさは 8ビットでなければなりません。

情報: USART RXを許可する必要はありません。デバイスはコンピュータからのデータが必要ではありません。USART TXだけが許可 されることを必要とします。 結果: Resource Management [MCC](資源管理)⇒Project Rersources(プロジェクト資源)⇒Peripherals(周辺機能)へ行き、USAR T1を選んでください。Easy Setup(容易な設定)タブで以下の構成設定が行われなければなりません。

- ・ Hardware Settings(ハートウェア設定)
  - Mode(動作形態)
  - Baud Rate(ホーレート)
  - Enable USART Transmitter(USART送信部許可): チェック
  - Parity Mode(ハ゜リティ動作)
  - Stop Bit Mode(停止ビット動作)
  - Character Size(文字の大きさ)

この設定は図2-18.で提示されます。

- : Async Mode(非同期動作)
- : 9600
- - : No Parity(ハ゜リティなし)
  - :1 stop bit(1停止ビット)
  - : 8 bit(8ビット)

<ul> <li>Hardware Settings</li> </ul>		
Mode:	Async Mode	•
Baud Rate:	1 ≤ 9600 ≤ 1000000	
Error Percent:	-0.032%	
8 Enable USART Receiver:		
8 Enable USART Transmitter:	$\checkmark$	
Parity Mode:	No Parity	•
Stop Bit Mode:	1 stop bit	•
Oharacter Size:	Character size: 8 bit	•

### 2.4.1.5. ピン単位部構成設定

Pin Manager: Grid View(ピン管理部:格子状表示)ウィンドウでADC用の入力ピンが選ばれなければなりません。PD3とPD4を図2-19.で描 かれるように選んでください。USART用の受信と送信のピンが下で提示されるように構成設定されなければなりません。

赵Z-19. C	ノ官理司	:恰丁ひ	、 <b>衣</b>	<b>(</b> 小	<																																						
Package:	QFN48 👻	Pin No:	44	45	46	47	48	1	2	3	4	5	6	7	8	9	10	11	12	13	16	17	18	19	20	21	22	23	24	25	26	27	30	31	32	33	34	35	36	37	38	39	40
						Por	t A	•					Port	в		_			F	Port	C V	,					F	ort	D	•		_	1	Por	t E '	,			Po	ort F	▼		_
Module	Function	Direction	0	1	2	3	4	5	6	7	0	1	2	3	4	5	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	0	1	2	3	4	5	6
ADC0	AINx	input																							ĵ.	î.	î.	ô	â	î.	î.	ĵ,	î.	ì	î.	î.	ì	î.	î.	î.	î.	î.	
Din Madula T	GPIO	input	î	î.	î	ĥ	î	î	ĵ.	î	î	î	î	î	î	î	î	ĵ.	î	î	ĵa	î	î	î.	ĵ.	î.	î.	ĵ.	î	ĥ	ì	ĵ,	î.	ĥ	î	î	î	î	î	î.	ì	î.	î.
Pin Module V	GPIO	output	î.	î.	ì	ĥ	î.	î.	î.	ĥ	î.	ì	Ъ	î.	î.	î.	Ъ	î.	î.	ì.	ĵ,	î.	î.	î.	î.	Ъ	î.	ĵ,	î.	Ъ	î.	ĵ,	Ъ	Ъ	î.	î.	Ъ	î.	î.	Ъ	î.	î.	î.
RSTCTRL	RESET	input																																									î.
	RXD	in/out																ô				ì																					
	TXD	output															Ô				ì																						

ADC入力として使うため、入力ピンはアナログ入力として構成設定されなければなりません(デジタル緩衝部が禁止されなければなりませ ん)。弱いプルアップと割り込みは禁止されなければなりません。

行うこと: MCCを使って以下の設定を行ってください。

- 各々のピンでデジタル入力緩衝部を禁止してください。
- ・各々のピンで弱いプルアップを禁止してください。
- ・割り込みを禁止してください。

結果: Resource Management [MCC](資源管理[MCC])⇒Project Rersources(プロジェクト資源)⇒Pin Module(ピン単位部)へ行 き、図2-20.で提示される構成設定を行ってください。

#### 図2-20. ピン単位部

Pin Name	Module	Function	Custom Name	OUTPUT	START HIGH	INVEN	PULLUPEN	ISC
PC0	USART1	TXD		$\checkmark$				Interrupt disabled but input buffer enabled
PC1	USART1	RXD						Interrupt disabled but input buffer enabled
PD4	ADC0	AINx						Digital Input Buffer disabled 🔹
PD3	ADC0	AINx						Digital Input Buffer disabled

### 2.4.2. 生成されたコート 概要

MCCで設計されたコードを生成するにはGenerate(生成)釦をクリックしてください。その後、応用開発 を続けるためMCCを閉じてください。MCCが生成したファイルは生成したプロジェクト下のProjects(プロ ジェクト)タブで見ることができます。



生成されたファイルは主ソースファイルでインクルードされ、システムが初期化されます。main.c用に生成されたコードが下で提示されます。

#include "mcc generated files/mcc h"	
<pre>int main(void) {</pre>	
/* MCU、ドライバ、ミドルウェアを初期化 */ SYSTEM_Initialize();	
/* あなたの応用コートで置き換えてください。 */ while (1)	
{ }	
} /**	
ファイルの終わり */	

SYSTEM\_Initialize関数はシステムと全ての周辺機能単位部(システム クロック、ADC、USART、VREF)を以前にMCCを使って構成設定したように初期化するのに使われます。

情報: ADC0、USART1、VREFを使って算法を実装するのに必要とされる全ての関数がMCCによって既に生成されています。

生成されたファイルは算法を実装するのに使用者を手助けするADCとUSARTの関数を提供します。この応用にとって関心のある関数はADC0\_GetDiffConversionとUSART1\_Writeです。MMCが生成したファイルでのそれらの実装が下のコード一覧で提示されます。

#### 例2-2. コード一覧2 - 差動変換結果を得るのに使われるADC関数

```
diff_adc_result_t ADC0_GetDiffConversion(adc_0_channel_t channel, adc_0_muxneg_channel_t channel1)
{
    diff_adc_result_t res;
    ADC0_StartDiffConversion(channel, channel1);
    while (!ADC0_IsConversionDone());
    res = ADC0_GetConversionResult();
    ADC0.INTFLAGS |= ADC_RESRDY_bm;
    return res;
}
```

```
例2-3.コードー覧3 - ADC結果を送信するのに使われるUSART関数
void USART1_Write(const uint8_t data)
{
    while (!(USART1.STATUS & USART_DREIF_bm));
    USART1.TXDATAL = data;
```

### 2.4.3. 設計された算法の実装

この副項は生成されたファイルを使ってmainファイルに算法を実装する方法を記述します。データ可視器(Data Visualizer)を使って応用を 試験するのに必要とされる関数も提供されます。

MPLABデータ可視器は書かれた組み込みデバイスとコンピュータ間の簡単な一方向通信を提供します。組み込み応用はデータを送信する 方法と時を制御しなければなりません。この例について、データは図2-22.で描かれるようにハプケットにされます。

図2-22. データ可視器ADCデータ形式

7レーム開始通票(1ハイト) ADC 12ビット結果下位(1ハイト) ADC 12ビット結果上位(1ハイト) 7レーム終了通票(1ハイト)

送信されたデータは開始と終了の通票によってフレーム化されます。これらはお互いに反転/1の補数です。データ可視器はフレーム通票と 本体の大きさで同期します。複数バイト変数では下位側バイトが先に送られなければなりません。

▲ 行うこと: ADC差動の結果を読んでコンピュータに送信するにはmainファイルに以下のコードが実装されなければなりません。

#### 例2-4. コード一覧4 - ADC結果の読み込みと送信

```
int main(void)
```

diff\_adc\_result\_t adcVal\_12b;

```
/* MCU、ドライバ、ミドルウェアを初期化 */
SYSTEM_Initialize();
```

while (1)

{

```
adcVa1_12b = ADC0_GetDiffConversion(ADC_MUXPOS_AIN3_gc, ADC_MUXNEG_AIN4_gc);
```

```
USART1_Write(START_TOKEN);
USART1_Write(adcVal_12b & 0x00FF);
USART1_Write(adcVal_12b >> 8);
USART1_Write(END_TOKEN);
```

「うっこと: START\_TOKEN(開始通票)とEND\_TOKEN(終了通票)は下で提示されるようにmain関数で使われるべくmain.cファイル でも定義されなければならない使用者定義されたマクロです。

#define START\_TOKEN 0x03
#define END\_TOKEN 0xFC

/\* フレーム開始通票 \*/ /\* フレーム終了通票 \*/



## 2.4.4. 応用試験 : デバッグ

応用の機能を試験するための1つの方法はこれを通して段階的に行い、全ての実装した命令が意図した結果を提供することによります。



行うこと: main関数内のSYSTEM\_Initialize();関数呼び出しで中断点(ブレークポイント)を許可してください。デ゙ベッグ動作へ移行してください。

2 結果: ファイル行傍らのエディタ側面帯でクリックすることによって新しい行中断点を作成してください。各々の行は下で提示されるように現れるでしょう。

#### 図2-23. 中断点許可



レシ、スタの内容を可視化するにはWindow(ウィント・ウ)⇒Debugging(デ・ハック)へ行き、IO View(I/O表示)を選んでください。これはIO View (I/O表示)ウィント・ウを開きます。デ・ハック・を開始するにはDebug(デ・ハック)へ行き、Debug Main Project(主プロジェクトをデ・ハック)を選んでください。プログラム実行は中断点の行で停止します。

Step Over(外側段階実行)釦クリック後、使用者は例えば、IO View(I/O表示)ウィンドウでレジスタを見ることによってADC周辺機能が望む ように初期化されたかを調べることができます。IO View(I/O表示)ウィンドウでのレジスタは図2-24.で提示されます。

🌈 行うこと: 全ての構成設定が望むように行われたかを調べてください。

図2-24. A	DC I/O表	<del>.</del>				
Icon P	eripheral					Option
Ac     Ac	DCO) esolution sele ccumulation s ock Pre-scale itial Delay Sel /indow Comp nalog Channe nalog Channe	ction (CTR Samples (C er (CTRLC) ection (CT varator Mo el Selection el Selection	RLA) TRLB) RLD) de (CTR Bits (MU Bits (MU	LE) JXNEG) JXPOS)		0x0 - 12-bit mode 0x0 - No accumulation 0x1 - CLK_PER divided by 4 0x0 - Delay 0 CLK_ADC cycles 0x0 - No Window Comparison 0x4 - ADC input pin 4 0x3 - ADC input pin 3
Icon	Name	Address	Value	Decimal	Bits	5
ADC DAC	(ADC0)					
😐 - 🗎	COMMAND	0x060A	0x00	0		10
<u> </u>	CTRLA	0x0600	0x21	33	0	543210
🗈	RUNSTBY		0	0	7	
B	CONVMODE		1	1		5
	LEFTADJ		0	0		4
	RESSEL		0	0		32
🖻	FREERUN		0	0		1
DAC	ENABLE		1	1		<b>0</b>
	CTRLB	0x0601	0x00	0		210
	CTRLC	0x0602	0x01	1		3210
O	PRESC		1	1		3210
	CTRLD	0x0603	0x00	0	2	
	CTRLE	0x0604	0x00	0		210
	DBGCTRL	0x060E	0x00	0		<u> </u>
	EVCTRL	0x060B	0x00	0		0
	INTCIRL	0x060C	0x00	0		10
	INTELAGS	0X060D	0000	0	_	
	MUXNEG	0X0609	0X04	4		
DAC	MUXNEG	0000	4	4		
	MUXPOS	0X0608	0x03	3		
DAC	MUXPOS		3	3		99432 <mark>1</mark> 0

応用を通して段階的に行うことにより、使用者は以下を知ることができます。

- ・ADC結果(<u>RES</u>)レジスタの変換結果
- ・TXDATALレジスタで USARTを通して送信されることが必要なデータ
- ・ Variables(変数)ウィンドウを使ってadcVal変数の値

### 2.4.5. 応用概要

応用はMPLABデータ可視器(Data Visualizer)を使うことによって容易に試験することができます。USARTを通して送信されたADC結果 を作図するには以下の手順が実行されなければなりません。

- 1. データ可視器を開いて図2-25. で提示されるようにプラ グインアイコンをクリックしてください。 図2-25. データ可視器を開く
  - Դ 🧝 🜔 👱 💁 🔂 🚯 -MCC PC:0x1FC ithsvnzc 🖭 main.c × 🖭 mcc.c × 🖭 adc0.c × Data Visualizer - 🖌 🎖 🗞 정 1월 🗐 🔵 🔲 🖉 📑 📴 Source History 👕 📴 🐼 - 💭 - 🔍 🖓 🖓 🖶 33 🖵 { 34 diff add result t addVal 12b; 35 /\* Initializes MCU, drivers and middleware \*/ 36 SYSTEM Initialize(); 38
- 2. 図2-26.で提示されるようにConnections(接続)タブでCuriosity Nano communic ation port (COMn)(Curiosity Nano通信ホート(COM n))引き落とし一覧を選ん でください。







4. Variable streamer Name(変数流し名)を選び、受信される変数を追加し、Next(次へ)をクリックしてください。

2-28. 変数流し構成	<u> </u>		
Plot Streaming D	ata from COM6		
Configure Variable Stre	amer		
j			
Variable Streamer Name:	ADC_Differential_results		
Framing Mode:	Auto 🔻		
Frame Size: (Including framing)	4 bytes		
Variable	Туре	Byte Position (Frame header is at position 0)	÷
ADC Result	Int16 💌	1	1 🗇
⊥ Import			Previous Nex

5. 望む変数を選ぶことによって作図される変数を選び、データを作図する方法(How to Plot)に対してNew axis per data type (1)(デー タ型毎に新しい軸)を選んでください。その後にFinish(終了)をクリックしてください。

Plot Streaming D	ata from COM6	×
Choose Variables to Pl	ot	
Variable Streamer:	ADC_Differential_results	
Variables to Plot:	All Variables	
	ADC Result: Int16	
How to Plot:	• New axis per data type (1)	
	New axis per variable (1)	
	Add 1 plots to selected axis	
	Add plots manually	
		Previous Finish

無作為に両可変抵抗器を回した後に作図が得られました。このADC結果が図2-30.で提示されます。



# <mark>3</mark>. 結び

この文書によって提供される訓練を経験した後、使用者はADCの基本的な機能を理解し、組み込み応用を開発するのに必要とされるソフトウェア ツールを使い、ADC単位部を使って基本的な応用を単独で開発することができるでしょう。更に、使用者は差動入力からデー タを変換するためのADC構成設定方法、データを継続的に変換する方法、そして結果を解釈する方法を理解するでしょう。この訓練は応用をデベッグして結果を可視化するのに必要な手順も提供します。

# 4. 参考文献

- 1. データ可視器(Data Visualizer)ソフトウェア使用者の手引き
- 2. MPLAB<sup>®コート</sup>構成部(MCC)
- 3. MPLABコード構成部(MCC)でのライブラリ追加方法
- 4. MCC最新版をダウンロードするためのMicrochipウェブ頁
- 5. MPLABデータ可視器での可視化デバッグ
- 6. 差動とシングル エントのADC白書
- 7. AVR128DA48 Curiosity Nano評価キット
- 8. Click boards™用Curiosity Nano Base

# 5. 改訂履歴

資料改訂	日付	注釈
А	2020年8月	初版文書公開

# Microchipウェフ゛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
- ・Microship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、その製品系統が安全であると考えます。
- ・Microchipデバイスのコート、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。これらのコート、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はそれ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います。
- ・Microchipや他のどの半導体製造業者もそれ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

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www.microchip.com/qualityを訪ねてください。

日本語© HERO 2020.

本訓練の手引きはMicrochipの訓練の手引き(DS40002244A-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: www.microchip.com/support ウェブ アトレス: 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 Fax: 905-695-2078

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

## 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 – 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

#### 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: 45-4485-5910 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 – カルヒング 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 ポーラント゛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