Microchip Nano デバッガ手引書

Nanoデバッガ

製品頁リンク

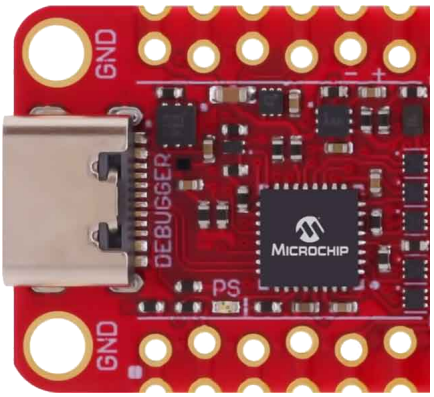
序文

Microchip Nanoﾃﾞﾊﾞｯｶﾞ はMicrochipと様々な協力社によって作られた入門向け開発ｷｯﾄで使われる費用対効果が高い書き込み器/ﾃﾞﾊﾞｯｶﾞ解決策です。それらの最も注目に値するのがCuriosity Nano基盤です。

Nanoﾃﾞﾊﾞｯｶﾞ は空間と費用が制限された開発基板とﾂｰﾙで必要とされる簡潔性と多用途性の独特な調和を提供します。

この文書について

このNanoﾃﾞﾊﾞｯｶﾞ手引書はMicrochipと協力社の様々な開発基板で見つけることができるNanoﾃﾞﾊﾞｯｶﾞに対する一般的な使用者の手引きです。Nonoﾃﾞﾊﾞｯｶﾞを利用するﾃﾞﾊﾞｯｶﾞや開発基板を使う場合、この文書はその製品の使用者手引きのﾃﾞﾊﾞｯｶﾞ固有部分を補完または置換します。



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

1. 使用事例

Nanoﾃﾞﾊﾞｯｶﾞは多くの方法で使うことができます。ここはいくつかの例です。

Curiosity Nano MCU基板

Curiosity Nano MCU基板はMicrochipからの新しいｼﾘｺﾝ製品を評価する一様な方法を提供する入門向け開発基板です。基板のﾋﾟﾝ配置は論理的で系統的な規則で標準化されてﾃﾞﾊﾞｲｽのﾋﾟﾝ配置に割り当てられます。これは全てのMCU基板に渡って周辺機能と機能を提供するための基本基板の組の使用を可能にします。

全てのCuriosity Nanoｷｯﾄは Nanoﾃﾞﾊﾞｯｶﾞを含みます。

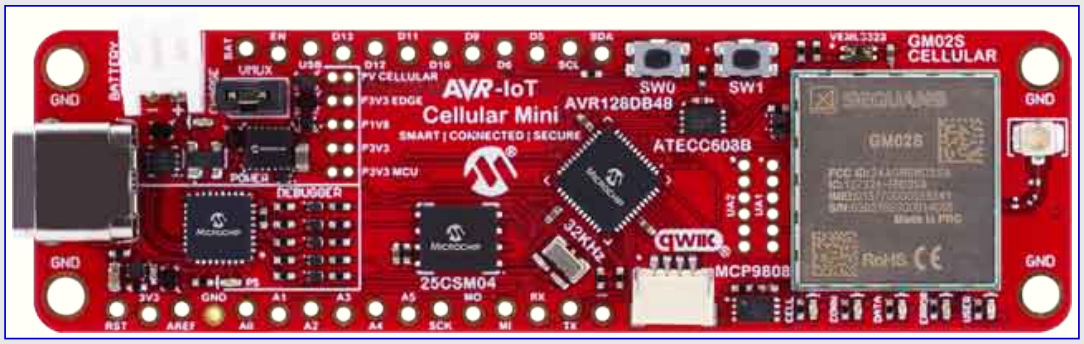
図1-1. PIC32CM PL10 Curiosity Nano

独自開発ｷｯﾄ

NanoﾃﾞﾊﾞｯｶﾞはARM®またはAVR®のﾏｲｸﾛ ｺﾝﾄﾛｰﾗを持つどの開発ｷｯﾄでも使うことができます。

例はAVR-IoT Cellular Miniです。

図1-2. AVR-IoT Cellular Mini



助言:「Nanoﾃﾞﾊﾞｯｶﾞ統合の手引き」に従うことによってNonoﾃﾞﾊﾞｯｶﾞを使ってあなた自身の開発ｷｯﾄを作ることができます。

独立型ﾃﾞﾊﾞｯｶﾞ作成

開発ｷｯﾄに実装される時にNanoﾃﾞﾊﾞｯｶﾞはﾃﾞﾊﾞｯｶﾞと基板制御機能の両方を提供します。独立型実装ではNanoﾃﾞﾊﾞｯｶﾞは定常的に基板に実装されないため、もはやその環境の完全な制御を持たず、目的対象ﾃﾞﾊﾞｲｽの情報も持ちません。この構成でのNanoﾃﾞﾊﾞｯｶﾞは任意選択の仮想ｼﾘｱﾙ ﾎﾟｰﾄ橋渡しと共に費用対効果が高いﾃﾞﾊﾞｯｶﾞです。

助言:「Nanoﾃﾞﾊﾞｯｶﾞ統合の手引き」に従うことによってNonoﾃﾞﾊﾞｯｶﾞを使ってあなた自身の独立型ﾃﾞﾊﾞｯｶﾞを作ることができます。

2. 能力

2.1. 特徴

鍵となるNanoデバッガの機能は次のとおりです。

- 工業標準のCMSIS-DAPインターフェースを経由するARM Cortexデバイスの書き込みとデバッグ
- CMSIS-DAPインターフェース供給業者拡張を経由するAVRと選ばれたPIC16, PIC18, PIC24, dsPIC32の書き込みとデバッグ
- 標準仮想シリアル ポート (CDC)
- キット情報読み込みと選ばれたデバイス系統のドラッグ&ドロップ書き込み用の大容量記憶実装
- 簡単な論理分析機能と基板制御機能用のデータ中継器インターフェース
- 多くのIDEと収益活動協調体制で支援される開放規約実装
- SAMD21 MCUを使い独自ハードウェア解決策へ統合するのに利用可能なファームウェア

2.2. 目的対象デバイス

ARM Cortexデバイスの書き込みとデバッグ

Microchip Nanoデバッガは ARMによって指定された標準CMSIS-DAPインターフェースを実装します。これはARM Cortexに基づくどのデバイスも本質的に書き込んでデバッグする能力があることを意味します。

CMSIS-DAP実装は第1版で、ホストPCとの通信にHIDインターフェースを使います。

SWO追跡は未だ支援されていません。

助言: 将来のファームウェア更新で、Nanoデバッガは大量/供給業者インターフェースを使うCMSIS-DAP第2版を支援します。旧来のIDE支援(例えば、Micoship Studio)についてはより古いファームウェアが使われなければなりません。

表2-1. ARM SWDピン配置

目的対象信号	目的	デバッガ ピン
SWDIO	直列線データ	DBG0
SWCLK	直列線クロック	DBG1
$\overline{\text{RESET}}$	目的対象リセット	DBG3

AVRデバイスの書き込みとデバッグ

Microchip Nanoデバッガは統一プログラム/デバッグ インターフェース(UPDI)を持つどのAVRデバイスとも使うことができます。このインターフェースは追加信号として$\overline{\text{RESET}}$を必要とするいくつかの実装(UDPIヘッダ第2版)と共に、単線、非同期、UARTに基づく規約(第1版)です。

注意 JTAG, デバッグWIRE, PDI, TPI, ISP, HVSP, HVPPを使う古いAVRデバイスはNanoデバッガによって支援されません。

UPDIインターフェースを使う不揮発性メモリの書き込みは当該AVRデバイス用データシートで文書化される一方で、デバッグ インターフェースは公開されていません。

AVRの書き込みとデバッグはCMSIS-DAP仕様で供給業者命令を使って実装されます。この規約で使われる命令一式は「EDBGに基づくツール規約」で文書化され、pyedbglibのコードで実装されます。

助言: Nanoデバッガはいくつかの改良と拡張を伴うEDBGの進化的拡張で、ほぼ同じ規約に従います。

表2-2. AVR UPDIピン配置

目的対象信号	目的	デバッガ ピン
UPDI	書き込みとデバッグ	DBG0
$\overline{\text{RESET}}$	(該当する場合、)目的対象リセット	DBG3

PIC®デバイスの書き込みとデバッグ

Microchip Nanoデバッガは下で一覧にしたデバイスの品種でだけ使うことができます。これらのデバイスの全てはMicrochipのICSP物理インターフェースを持ちますが、デバイス系統間で論理的に異なります。これは以下を含みます。

- PIC16デバイス
- PIC18デバイス
- PIC24デバイス
- dsPIC32デバイス

NanoﾃﾞﾊﾞｯｶﾞはMPLAB® PICkit™ 5(と同様の)ツールを使ってPICﾃﾞﾊﾞｲｽの書き込みとﾃﾞﾊﾞｯｸﾞに使われるのと非常に似たｽｸﾘﾌﾟﾄ ｴﾝｼﾞﾝを含みます。支援されるﾃﾞﾊﾞｲｽ用のﾃﾞﾊﾞｲｽ系統一括のscriptsﾌｫﾙﾀﾞにPyythonｽｸﾘﾌﾟﾄが含まれます。Python階層は書き込み算法をｽｸﾘﾌﾟﾄ ｴﾝｼﾞﾝ用のﾊﾞｲﾄ ｺｰﾄﾞに変換し、これはCMSIS-DAPｲﾝﾀｰﾌｪｰｽで同じ供給業者命令を使ってNanoﾃﾞﾊﾞｯｶﾞに渡されます。

重要: PICの書き込みｲﾝﾀｰﾌｪｰｽはﾃﾞﾊﾞｲｽ書き込み仕様で文書化されますが、ﾃﾞﾊﾞｯｸﾞ規約は公開されていません。書き込みとﾃﾞﾊﾞｯｸﾞに使うｽｸﾘﾌﾟﾄ言語は独占で公開されていません。

表2-3. PIC ICSPﾋﾟﾝ配置

目的対象信号	目的	ﾃﾞﾊﾞｯｶﾞ ﾋﾟﾝ
ICSPDAT	書き込みﾃﾞｰﾀ	DBG0
ICSPCLK	書き込みｸﾛｯｸ	DBG1
$\overline{\text{MCLR}}$	主解消(ﾘｾｯﾄ)	DBG3

3. NanoﾃﾞﾊﾞｯｶﾞUSBインターフェース

Nanoﾃﾞﾊﾞｯｶﾞはホスト コンピュータに対して次のような4つのインターフェースを持つ複合USB装置として現れます。

- CMSIS-DAP – Cortex MCUの書き込みとﾃﾞﾊﾞｯｸﾞのためにARMによって提供された工業標準開放仕様
- 仮想シリアル ポート (CDC) – 標準シリアル ポート インターフェース
- 大容量記憶クラス装置 (MSC) – 標準取り外し可能記憶'ﾃﾞｨｽｸ'
- ﾃﾞｰﾀ中継器インターフェース – ﾃﾞｰﾀをホスト コンピュータへ流すためのMicrochip占有インターフェース

留意: ﾃﾞﾊﾞｯｶﾞのファームウェアを最新に保ってください。ファームウェア更新は現在のMicrochip IDEを使う時に自動的に行われます。またはPKOB nano支援用の更新されたツール一括を調べ、ファームウェアを手動で更新するのにpydebuggerupgradeを使ってください。

4. CMSIS-DAPデバッグ インターフェース

Nanoデバッガ はARMによって標準指定されたCMSIS-DAPデバッガ インターフェースの実装で、ホスト コンピュータのUSB下部組織で人間インターフェース装置(HID:Human Interface Device)として現れます。

HID装置はそれらがマウスとキーボードの応用に対して広く使われるため、一般的に特定のオペレーティング システム ドライバを必要としません。これはそれらがシステム構成やデバイス マネージャでMicrochip装置として当然現れ得ないことも意味します。

助言: デバイス ドライバについてのより多くの情報については「USBドライバ」項をご覧ください。

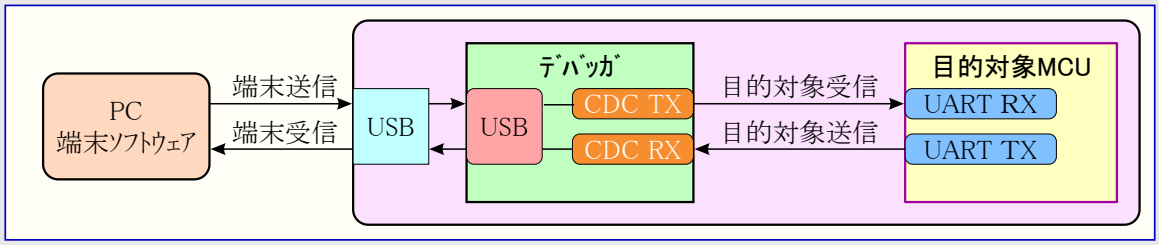
5. 仮想シリアル ポート (CDC)

仮想シリアル ポート(CDC)はホストPCと目的対象デバイス間の汎用シリアル橋渡しです。

5.1. 概要

Nanoデバッガはホストで仮想シリアル ポートとして現れる標準通信装置クラス(CDC:Communications Device Class)を持つ複合USB装置を実装します。ホスト コンピュータと目的対象間の両方向で任意データを流すのにCDCを使ってください。ホスト コンピュータで仮想シリアル ポートを通して送られた全ての文字はデバッガのCDC TXピンでUARTとして送られます。デバッガのCRC RXピンで捕獲されたUART文字は仮想シリアル ポートを通してホスト コンピュータに返されます。

図5-1. CDC接続



情報: 上図で示されるように、ホスト コンピュータから受け取る文字に対してデバッガのCDC TXピンは目的対象のUART RXピンに接続されます。同様に、ホスト コンピュータへ送られる文字に対してデバッガのCDC RXピンは目的対象のUART TXピンに接続されます。

5.2. オペレーティング システム支援

Windows®機ではCDCがCuriosoty Virtual COM Port(Curiosity仮想COMポート)として列挙(接続認識)され、Windowsデバイス マネージャのポート部分に現れます。COMポート番号はそこで見つけることもできます。

情報: 古いWindowsシステムではCDCがUSBドライバを必要とします。MPLAB® X IDEのインストールはこのドライバを含みます。

Linux®機ではCDCが/dev/ttyACM#として列挙(接続認識)されて現れます。

情報: Linuxでtty*装置は"dialout"群に属し、故にCDCアクセスする許可を持つ群の一員になることが必要かもしれません。

MAC®機ではCDCが/dev/tty.usbmodem#として列挙(接続認識)されて現れます。使う端末プログラムに依存し、usbmodem#として利用可能なモデムの一覧で現れます。

5.3. 制限

Nanoデバッガのcdcで全てのUART機能が実装される訳ではありません。制限は以下のようにここで概説されます。

- **ボーレート:** 1200bps～500kbpsの範囲でなければなりません。この範囲外のどのボーレートも警告なしに最も近い限度に設定されます。ボーレートは実行中に変えることができます。
- **文字形式:** 8ビット文字だけが支援されます。
- **パリティ:** 奇数、偶数、なしにすることができます。
- **ハードウェア流れ制御:** 支援なし
- **停止ビット:** 1または2のビットが支援されます。

5.4. 合図

USB列挙(接続認識)の間、ホストOSはCDCインターフェースの通信とデータのパイプを開始します。この時点で、CDCのボーレートと他のUARTパラメータを設定して読み戻すことが可能ですが、データの送出と受け取りは許可されません。

ホストに接続する時に端末はDTR信号を活性にしなければなりません。これがUSBインターフェースで実装される仮想制御信号のため、基板には物理的に存在しません。ホストからのDTR活性化はNanoデバッガにCDC作業が活性であることを示します。デバッガは(利用可能ならば)それの基準移転器(レベルシフタ)を許可してCDCデータの送受信機構を開始します。

デバッガ ファームウェア 1.20またそれ以前版でのDTR信号の不活性化は以下の動きを持ちます。

- デバッガUART受信部が禁止され、それ以上データはホスト コンピュータへ転送されません。
- デバッガUART送信部は転送用に準備し待ち行列にしたデータを送り続けますが、ホスト コンピュータからの新データを受け入れません。
- (利用可能なら)基準移転器は禁止されず、デバッガCDC TX線は駆動されたままに留まります。

デバッガ ファームウェア 1.21またそれ以降版でのDTR信号の不活性化は以下の動きを持ちます。

- デバッガUART受信部が禁止され、更なるデータはホスト コンピュータへ転送されません。

- デバッガUART送信部は転送用に準備し待ち行列にしたデータを送り続けますが、ホスト コンピュータからの新データを受け入れません。
- 一旦進行中の送信が完了すると、基準移転器が禁止され、故にデバッガCDC TX線は高インピーダンスになります。

留意: 端末模倣部をDTR信号有効に設定してください。この信号なしではNanoデバッガがそれのUARTを通すどのデータの送信も受信もしません。

助言: NanoデバッガのCDC TXピンはホストによってCDCインターフェースが許可されるまで駆動されません。また、デバッガと目的対象に接続しているCDC線上に外部プルアップ抵抗がなく、電源投入中にその線が浮くことを意味します。フレーミング異常などのような予測不能な動きに帰着する不具合も避けるため、目的対象デバイスはデバッガのCDC TX線に接続されたピンで内部プルアップを許可することができます。

5.5. 高度な使い方

CDC置き換え動作

普通の動作では、Nanoデバッガはホストとデバイス間のUART橋渡しです。けれども、或る使用事例で、Nanoデバッガは基本動作形態を置き換えて他の目的のためにCDC TXとRXのピンを使うことができます。

Nanoデバッガの大容量記憶ドライブへの文書ファイル引き摺りで、デバッガのCDC TXピンの出力に文字を送ることができます。ファイル名と拡張子は普通ですが、文書ファイルは次のような文字で始まります。

```
CMD:SEND_UART=
```

デバッガ ファームウェア 1.20またそれ以前版は以下の制限を持ちます。

- 最大メッセージ長は50文字で、フレーム内の全ての残りデータは無視されます。
- この動作で使われる既定ボーレートは9600bpsですが、CDCが既に活性、または構成設定されていた場合、以前に使われたボーレートが未だ適用されます。

デバッガ ファームウェア 1.21またそれ以降版は以下の制限/機能を持ちます。

- 最大メッセージ長はホスト コンピュータとオペレーティング システムでのMSC/SCSI層制限時間に依存して変わるでしょう。512バイトの単一SCSIフレーム(498文字の本体)が保証され、4Kバイトまでのファイルが殆どのシステムで動くでしょう。転送はファイルで出会った最初のNULL文字で完了されます。
- 使うボーレートは既定命令に対して常に9600bpsです。

  ```
  CMD:SEND_UART=
  ```

  CDC置き換え動作はCDC/端末上でのデータ転送と同時に使わないでください。ファイルがCDC置き換え動作経由で受信している時にCDC端末作業が活動の場合、その操作の間一時休止され、一旦完了すると再開されます。
- 明示的なボーレートで以下のような追加命令が支援されます。

  ```
  CMD:SEND_9600=
  CMD:SEND_115200=
  CMD:SEND_460800=
  ```

USB段階のフレームの考慮

ホストからCDCへ送るデータはバイト単位、または64バイトUSBフレーム内に切り分けられる塊で行うことができます。このような各々のフレームはデバッガのCDC TXピンへ送るため、待ち行列にされます。フレーム毎に少量のデータを送ると、Nanoデバッガがバイトではなくフレームを緩衝するため、特に低ボーレートで非効率になり得ます。最大4つの64バイト フレームを何時でも活性にすることができます。Nanoデバッガはそれによってやって来るフレームを調整します。データを含む完全な64バイト フレームの送信が最も効率的な方法です。

デバッガのCDC RXピンでデータを受け取る時に、Nanoデバッガはやって来るバイトを64バイト フレームへ一列に並べ、それらが満たされた時にホストへ送るためにUSB待ち行列に送られます。不完全なフレームも概ね100ms間隔でUSB待ち行列へ押し込まれ、USBフレーム開始通票によって起動されます。何時でも最大8つの64バイト フレームを活性にすることができます。

ホスト(または走行しているソフトウェア)が充分速くデータを受け取ることに失敗した場合、オーバーランが発生します。これが起きると、USB待ち行列に送られつつあるものに代わって最後に満たされた緩衝部フレームが再使用され、完全なフレーム データが失われます。これを防ぐため、使用者はCDCデータ パイプが継続的に読まれるか、または代わりにやって来るデータ速度を減らすかを保証しなければなりません。

中断(Break)文字送出

ホストはCDCを使う装置にUART中断文字を送ることができ、これは受信部の状態機構をリセットしたり、装置で走行する応用にホストから例外条件を合図するのに使えます。

中断文字はホストから装置へ送られる最低11ビットの'0'の連続です。

全てのUART受信部が中断を検出するための支援を持つ訳ではありませんが、正しい形式の中断文字は通常、受信部でフレーミング異常を起動します。

ﾃﾞﾊﾞｯｶﾞのCDCを使う中断文字送信は以下の制限を持ちます。

- 中断はCDC置き換え動作(ﾄﾞﾗｯｸﾞ&ﾄﾞﾛｯﾌﾟ)と同時に送ってはなりません。これら両機能は(限られた時間続く)一時的な状態で、独立して使われなければなりません。
- 中断送出は送られつつあるどのﾃﾞｰﾀも損失させます。中断を送る前に送信緩衝部内の全ての文字が送られるのを許すために充分な量の時間待つのを確実にしてください。これは予期した中断文字の使い方に沿っていてもで、例えば、制限時間後の受信部状態機構ﾘｾｯﾄはﾎｽﾄに返されるﾃﾞｰﾀを待たせます。
- CDC仕様は最大65534msの持続時間のﾃﾞﾊﾞｯｶﾞ計時の中断の要求を許します。簡単にするため、ﾃﾞﾊﾞｯｶﾞは中断持続時間を支援する最低ﾎﾞｰﾚｰﾄで最大11ﾋﾞｯﾄ持続に制限します。
- CDC仕様は不定のﾎｽﾄ計時中断を許します。この場合、中断状態を開放するのは端末応用または使用者の責任です。

注: 中断文字送出はﾃﾞﾊﾞｯｶﾞ ﾌｧｰﾑｳｪｱ1.24とそれ以降版で利用可能です。

6. 大容量記憶装置

Nanoﾃﾞﾊﾞｯｶﾞはそれが接続されるﾎｽﾄ ｵﾍﾟﾚｰﾃｨﾝｸﾞ ｼｽﾃﾑ経由で読み書き操作に対してｱｸｾｽができる簡単な大容量記憶装置実装を含みます、

これは以下を提供します。

- ｷｯﾄの情報と支援を詳述するための基本的な文書とHTMLのﾌｧｲﾙに対する読み込みｱｸｾｽ
- Intel® HEXとUF2の形式のﾌｧｲﾙを目的対象ﾃﾞﾊﾞｲｽのﾒﾓﾘに書くための書き込みｱｸｾｽ
- 有用な目的用の簡単な文書ﾌｧｲﾙのための書き込みｱｸｾｽ

注: UF2形式の支援はﾃﾞﾊﾞｯｶﾞ ﾌｧｰﾑｳｪｱ1.31またはそれ以降版で利用可能です。

6.1. 大容量記憶装置実装

Nanoﾃﾞﾊﾞｯｶﾞは部分的にFAT12それ自身の特質とそれの組み込み応用に対する目的を満たすための最適化のため、いくつかの制限を持ち高く最適化されたFAT12ﾌｧｲﾙ ｼｽﾃﾑの変種を実装します。

Curiosity Nano USB装置は大容量記憶装置としてUSB第9節適合ですが、汎用大容量記憶装置で期待するものを多少なりとも満たしません。この動きは意図的です。

Windowsｵﾍﾟﾚｰﾃｨﾝｸﾞ ｼｽﾃﾑ使用時、Nanoﾃﾞﾊﾞｯｶﾞはﾃﾞﾊﾞｲｽ ﾏﾈｰｼﾞｬのﾃﾞｨｽｸ部分で見つかるCuriosity Nano USB装置として列挙(認識)されます。CURIOSITYﾄﾞﾗｲﾌﾞはﾌｧｲﾙ ﾏﾈｰｼﾞｬに現れ、ｼｽﾃﾑで次に利用可能なﾄﾞﾗｲﾌﾞ文字を獲得します。

CURIOSITYﾄﾞﾗｲﾌﾞは概ね1Mﾊﾞｲﾄの空き空間を含み、目的対象ﾃﾞﾊﾞｲｽのﾌﾗｯｼｭ ﾒﾓﾘの大きさを反映しません。Intel HEXまたはUF2のﾌｧｲﾙ書き込み時、重大な付随負荷を与える付加ﾃﾞｰﾀを持つASCIIで符号化され、故に1Mﾊﾞｲﾄはﾃﾞｨｽｸの大きさ用に無作為に選ばれた値です。

CURIOSITYﾄﾞﾗｲﾌﾞをﾌｫｰﾏｯﾄすることは不能です。目的対象へのﾌｧｲﾙ書き込み時、ﾌｧｲﾙ名がﾃﾞｨｽｸ ﾃﾞｨﾚｸﾄﾘ一覧に現れるかもしれませんが、これは単にｵﾍﾟﾚｰﾃｨﾝｸﾞ ｼｽﾃﾑのﾃﾞｨﾚｸﾄﾘ表示にすぎず、現実には更新されません。そのﾌｧｲﾙ内容を読み出すことは不可能です。基板を取り外して再接続すると、ﾌｧｲﾙ ｼｽﾃﾑをそれの元の状態に戻しますが、目的対象は未だ直前に書かれた応用を含みます。

目的対象ﾃﾞﾊﾞｲｽを消去するにはﾃﾞｨｽｸに"CMD:ERASE"で始まる文字ﾌｧｲﾙを複写してください。

既定でCURIOSITYﾄﾞﾗｲﾌﾞはｱｲｺﾝ生成、状態報告、更なる情報へのﾘﾝｸに使われる以下のようないくつかの読み込み専用ﾌｧｲﾙを含みます。

- AUTORUN.ICO – Microchipﾛｺﾞ用ｱｲｺﾝ ﾌｧｲﾙ
- AUTORUN.INF – ｱｲｺﾝ ﾌｧｲﾙを表示するためにWindowsのｴｸｽﾌﾟﾛｰﾗに対して必要とされるｼｽﾃﾑ ﾌｧｲﾙ
- CLICK-ME.HTM – ｷｯﾄ固有ｳｪﾌﾞ実演応用へのﾘﾀﾞｲﾚｸﾄ
- KIT-INFO.HTM – 開発基板ｳｪﾌﾞｻｲﾄへの向け直し
- KIT-INFO.TXT – 基板のﾃﾞﾊﾞｯｶﾞ ﾌｧｰﾑｳｪｱ版、基板名、USB通番、ﾃﾞﾊﾞｲｽ、ﾄﾞﾗｯｸﾞ&ﾄﾞﾛｯﾌﾟ支援の詳細を含む文字ﾌｧｲﾙ
- PUBKEY.TXT – ﾃﾞｰﾀ暗号化用公開鍵を含む文字ﾌｧｲﾙ
- STATUS.TXT – 基板の書き込み状態を含む文字ﾌｧｲﾙ

情報: NanoﾃﾞﾊﾞｯｶﾞはSTATUS.TXTを動的に更新します。けれども、ｵﾍﾟﾚｰﾃｨﾝｸﾞ ｼｽﾃﾑがﾌｧｲﾙをｷｬｯｼｭし得るため、内容が正しい状態を反映しないかもしれません。

助言: CLICK-MEとPUBKEYのﾌｧｲﾙの存在はﾃﾞﾊﾞｯｶﾞ基板構成に依存します。

6.2. ﾋｭｰｽﾞ/構成設定のﾊﾞｲﾄ/語

ﾋｭｰｽﾞ ﾊﾞｲﾄ (AVR® MCU目的対象)

ﾄﾞﾗｯｸﾞ&ﾄﾞﾛｯﾌﾟ書き込み実行時、Nanoﾃﾞﾊﾞｯｶﾞは統一ﾌﾟﾛｸﾞﾗﾑ/ﾃﾞﾊﾞｯｸﾞ ｲﾝﾀｰﾌｪｰｽ(UPDI)を禁止しようとするどのﾋｭｰｽﾞ ﾋﾞｯﾄも遮蔽し、これはUPDIﾋﾟﾝをﾘｾｯﾄまたはGPIO動作で使うことができないことを意味します。UPDIﾋﾟﾝの代替機能の1つの選択は外部ﾃﾞﾊﾞｯｶﾞの高電圧UPDI活性化の能力を使わないと、ﾃﾞﾊﾞｲｽをｱｸｾｽ不能にします。

構成設定ﾊﾞｲﾄ/語 (PIC® MCU目的対象)

ﾄﾞﾗｯｸﾞ&ﾄﾞﾛｯﾌﾟ書き込み実行時、ﾃﾞﾊﾞｯｶﾞはICSPｲﾝﾀｰﾌｪｰｽ、または他のﾜﾝﾀｲﾑ書き込み(OTP)構成設定を禁止しようとするどの構成設定ﾊﾞｲﾄ/語も遮蔽します。

6.3. ドラッグ&ドロップ書き込みの制限

ARM Cortexデバイス

ドラッグ&ドロップ書き込みはARM Cortexデバイスで支援されません。書き込みについては支援するIDEまたはコマンド行ユーティリティの1つを使ってください。

施錠ビット

Hexファイルに含まれる施錠ビットはドラッグ&ドロップ書き込みを使う時に無視されます。施錠ビットを書くには支援するIDEの1つを使ってください。

ヒューズでのCRC検査許可

ドラッグ&ドロップ書き込みを使う時にデバイスのヒューズでCRC検査を許可することはお勧めできません。これは(ヒューズ ビットに影響を及ぼさない)後続するチップ消去がCRC不整合を起こし、応用が起動に失敗するからです。支援するIDEの1つを使ってチップ消去が実行されなければならず、この状態から目的対象を回復するため、消去後、自動的にCRCヒューズを解消します。

6.4. 特殊命令

大容量記憶装置への文字ファイル複写によっていくつかの有用な命令が支援されます。ファイル名と拡張子は無関係で、命令処理部は内容だけに反応します。

表6-1. 特殊ファイル命令

命令内容	説明
CMD:ERASE	目的対象チップ消去を実行
CMD:SEND_UART=	CDC UARTに文字列を送信。「高度な使い方」をご覧ください。
CMD:SEND_9600= CMD:SEND_115200 CMD:SEND_460800	指定したボーレートで文字列をCDC UARTへ送ります。ここで明示的に指定したボーレートだけが支援されることに注意してください!。「高度な使い方」をご覧ください。(デバッガ ファームウェア1.25版またはより新しい版)
CMD:RESET	書き込み動作へ入って直後に書き込み動作を抜け出すことによって目的対象をリセット。正確なタイミングは目的対象デバイスの書き込みインターフェースに従って変わり得ます。(デバッガ ファームウェア1.25版またはより新しい版)
CMD:POWERTOGGLE	目的対象を電源断して100ms遅延後に電源を回復します。外部電力が提供されている場合、この命令は無効です。(デバッガ ファームウェア1.25版またはより新しい版)
CMD:0V	目的対象供給調整器を禁止することによって目的対象デバイスを電源断。外部電力が提供されている場合、これは無効です。(デバッガ ファームウェア1.25版またはより新しい版)
CMD:1V8	目的対象電圧を1.8Vに設定。外部電力を使う場合、この命令は無効です。(デバッガ ファームウェア1.25版またはより新しい版)
CMD:3V3	目的対象電圧を3.3Vに設定。外部電力を使う場合、この命令は無効です。(デバッガ ファームウェア1.25版またはより新しい版)
CMD:5V0	目的対象電圧を5.0Vに設定。外部電力を使う場合、この命令は無効です。(デバッガ ファームウェア1.25版またはより新しい版)

情報: 模倣した大容量記憶ディスクに送られた内容は上の表で一覧にされる命令を起動し、成功と失敗のどちらの場合も反応は提供しません。

6.5. UF2形式を使うドラッグ&ドロップ書き込み

ドラッグ&ドロップ用UF2形式

ドラッグ&ドロップ書き込みは開発キット上のマイクロ コントローラの不揮発性メモリを書くための簡単な仕組みを提供します。これは代表的に書式の一部として必要なアドレスとセグメントの情報を含むIntel® HEX形式を使って行われます。Intel HEXファイルはASCII文字として符号化したメモリ内容を含み、これは厳密な順番で解析されなければなりません。これはホスト オペレーティング システムが正しい順序で内容を送らなければならないことを意味します。これは全てのオペレーティング システムの全ての変種の場合に常に該当する訳ではありません。UF2形式は順序どおり以外で転送されるメモリを許す手段としてMicrosoftによって開発されました。これはデータ転送経路全体を通して固定の塊の大きさを強制することによって達成され、それによって部分的な書き込みを防ぎます。

UF2形式のより多くの情報はGitHubで利用可能です。

UF2ファイルの生成

プロジェクト コンパイル手順の結果は代表的にIntel HEXファイルで、これは構築後段階を使ってUF2ファイルに変換することができます。pypi.orgで配給されるpymcuprog一括は3.17またはそれ以降版でIntel HEXファイルをUF2ファイルに変換する関数を含みます。

助言: この手順はその環境にPyrhon 3とpymcuprog一括の最新公開版のインストールだけでなくPythonスクリプト フォルダがシステムまたはユーザのパスに含まれることが必要です。

Intel HEXファイルは次のようなpymcuprogコマンド行インターフェースを使ってUF2ファイルに変換することができます。

```
pymcuprog makeuf2 -f app.hex --uf2file app.uf2
```

この手順はMPLAB X IDEで構築後段階を追加することによって合理化することができます。Project Properties(プロジェクト特性)構成設定ダイアログでBuilding(構築)タブを選び、Execute this line after build(構築後この行を実行)をチェックしてください。編集枠に次のような指令を追加してください。

```
pymcuprog makeuf2 -f ${ImagePath} --uf2file ${ImageDir}¥${ProjectName}.X.${IMAGE_TYPE}.uf2
```

応用が構築される時毎に生成したIntel HEXファイルは同じフィイル名で.uf2ファイル拡張子を持つUF2ファイルへ自動的に変換されます。

7. データ中継器インターフェース (DGI)

データ中継器インターフェース(DGI:Data Gateway Interface)はNanoデバッガとホスト コンピュータに基づく可視化ツール間で生と時刻印されたデータを転送するUSBインターフェースです。ホスト コンピュータで何れのデバッグGPIOデータを表示するのにもMPLAB Data Visualizer(データ可視器)が使われます。これはMPLAB® X IDE用プラグインとしてまたはMPLAB® X IDEと並行して使うことができる独立型応用として利用可能です。

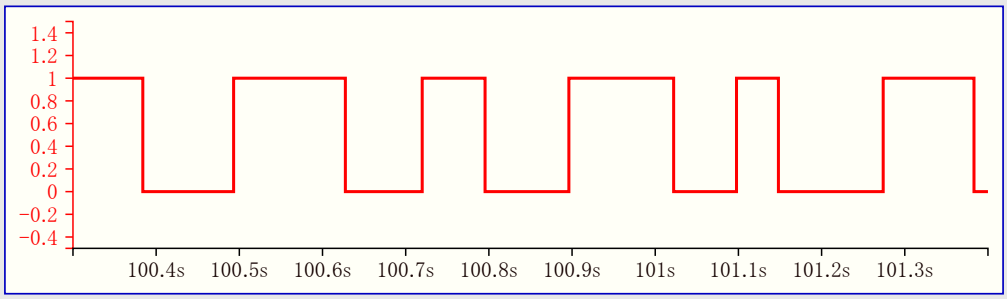
助言: GPIOのチャネル数はキットに実装されたデバイスに応じて変わり得ます。AVRデバイスが2つのチャネルを持つ一方で、他のデバイス型は単一チャネルを持ちます。

7.1. デバッグGPIO

デバッグGPIOチャネルは目的対象応用をホスト コンピュータ可視化応用に接続する時刻印されたデジタル信号線です。これらは例えば、特定の応用状態遷移が起きた時のような、代表的に時間軸での低周波数事象の発生を作図するのに使われます。

下図はMPLABデータ可視器(Data Visualizer)でデバッグGPIOに接続された機械的な切替器のデジタル状態の監視を示します。

図7-1. MPLABデータ可視器でのデバッグGPIO監視



デバッグGPIOチャネルは時刻印され、故にDGI GPIO事象の分解能はDGI時刻印単位部分解能によって決められます。

重要: より高い周波数の信号集中の捕獲が可能でも、GPIOが使える周波数範囲は最大約2kHzです。この周波数を超える信号を捕獲する試みはデータの飽和と溢れに帰着し、DGI作業の中断を引き起こすかもしれません。

7.2. 時刻印

デバッガによって捕獲される時にDGI供給元は時刻印されます。Curiosity Nanoデバッガで実装される時刻印計数器は0.5µsの時刻印分解能を提供する2MHzの周波数で増されます。

8. 基板制御器機能

8.1. 電圧監視部

Nanoﾃﾞﾊﾞｯｶﾞは目的対象ﾃﾞﾊﾞｲｽの電圧を監視するのにADCﾁｬﾈﾙを使います。目的対象電圧読み取りは基板構成設定のTVR(目的対象電圧読み取り)領域によって制御され、この機能ﾋﾞｯﾄが解除(0)される場合に電圧監視は禁止されます。

電圧監視部はNanoﾃﾞﾊﾞｯｶﾞ ﾌｧｰﾑｳｪｱで走行し、一定間隔でADCを採取する裏ﾀｽｸです。

基板上ﾃﾞﾊﾞｯｶﾞとして使われる時に、基板設計者は最小動作電圧値を指定する、基板のVMIN領域を設定します。この電圧未満で、電圧監視部は目的対象電圧がOFFであると記録します。OFF状態での書き込みとﾃﾞﾊﾞｯｸﾞの操作の試みの扱いは目的対象ﾃﾞﾊﾞｲｽ型に依存します。

現在の動作電圧はMicrochip IDEまたはpymcuprogを使って次のように読み出すことができます。

```
pymcuprog getvoltage
```

供給電圧に対する設定点を読むには次を使ってください。

```
pymcuprog getsupplyvoltage
```

USB電圧

Nanoﾃﾞﾊﾞｯｶﾞは電源であるUSB VBUS線の電圧を読むこともできます。これは主に利便性の機能です。

USB電圧はMicrochip IDEまたはpymcuprogを使って次のように読むことができます。

```
pymcuprog getusbvoltage
```

8.2. 電圧制御

Nanoﾃﾞﾊﾞｯｶﾞは自身の電力元で目的対象ﾃﾞﾊﾞｲｽに供給することができます。これは基板上と独立型ﾃﾞﾊﾞｯｶﾞの両構成設定に対して任意選択です。

動作電圧を変更するにはMicrochip IDEまたはpymcuprogの1つを次のように使ってください。

```
pymcuprog --setsupplyvoltage
```

助言: 基板にﾄﾞﾗｯｸﾞ&ﾄﾞﾛｯﾌﾟ命令文字ﾌｧｲﾙを複写することによって目的対象電圧を調節する容易な任意選択があり、これは一般的に使われる目的対象電圧一式を支援します。更なる詳細については「特殊命令」項をご覧ください。

目的対象電圧を制御するためにNanoﾃﾞﾊﾞｯｶﾞによって使われる機構は実施された基板構成設定に依存します。

表8-1. 電圧制御用基板構成設定

領域	説明	注釈
ANALOG_FEATURES: TVS	目的対象電圧設定	目的対象電圧を制御できることを示します。
VMIN	最小動作電圧	この値未満で機能しなくなる電圧を設定
VMAX	最大動作電圧	この値越えで機能しなくなる電圧を設定
VTG	既定動作電圧	電圧が設定されない時に使われる電圧
VREG	電圧調整器構成設定	調整器が基板上に実装されていることを示します。

電圧調整器高位性設定 0 – なし

この構成設定が使われる時に電圧調整器は利用できません。

電圧調整器高位性設定 1 – Curiosity Nano

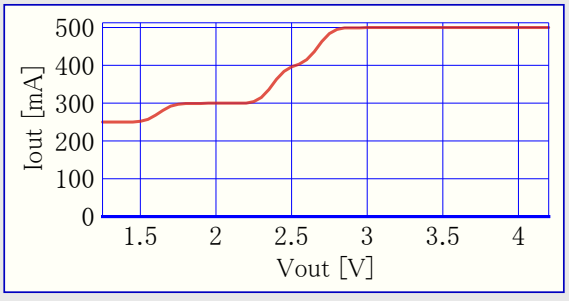
注: この構成設定はCuriosity Nano基板(調整可能な電圧基板)で使われます。

VREG基板構成設定が$01に設定されると、Nanoﾃﾞﾊﾞｯｶﾞは低損失調整器の帰還ﾋﾟﾝの電圧を操作することによって目的対象ﾃﾞﾊﾞｲｽの電圧を調整するのにDACを使います。Nanoﾃﾞﾊﾞｯｶﾞは「電圧監視部」で記述されるように調整器の出力を監視し、それによって電圧を調整します。

目的対象電圧調整器はMIC5353可変出力LDOです。NanoﾃﾞﾊﾞｯｶﾞはMIC5353の帰還電圧を操作することによって基板の目的対象部分へ供給する電圧出力を調節することができます。このﾊｰﾄﾞｳｪｱ実装は概ね1.7～5.1Vの電圧範囲に制限されます。

MIC5353は500mAの最大電流負荷を支援します。これは小さな印刷回路基板(PCB)上に配置された小さな外囲器のLDO調整器で、500mAより低い負荷で温度遮断条件に達し得ます。最大電流負荷は入力電圧、選んだ出力電圧、周囲温度に依存します。右図は5.1Vの入力電圧と23℃の周囲温度での調整器に対する安全動作領域を示します。

図8-1. 目的対象調整器安全動作領域



注意 ここで示した概略はCuriosity Nano基板で測定され、PCBに依存して変わるかもしれません。

目的対象調整器の電圧出力はNanoﾃﾞﾊﾞｯｶﾞによって継続的に監視(測定)されます。期待する電圧を100mV上回るまたは下回る変動が検出された場合に異常状態が起動されます。この場合、電圧調整器はOFFに切り換えられます。

注意 異常検出は外部的に印加された電圧での変動も扱います。より多くの情報については「電圧OFFﾋﾟﾝ (VOFF)」をご覧ください。

8.3. 電圧OFFﾋﾟﾝ (VOFF)

注意 VOFFﾋﾟﾝは調節可能な調整器が使われる時にだけ関連します。

基板上の目的対象調整器を使う代わりに、目的対象ﾃﾞﾊﾞｲｽに給電するのに外部電圧を使うことができます。電圧OFF(VOFF)ﾋﾟﾝが接地(GND)ﾋﾟﾝに短絡されると、Nanoﾃﾞﾊﾞｯｶﾞ ﾌｧｰﾑｳｪｱは目的対象調整器を禁止し、VTGﾋﾟﾝに外部電圧を安全に印加するようにします。

VOFFﾋﾟﾝは実装されたCriosity Nanoｷｯﾄに電力を提供することをCuriosity Nano基本基板に許すのに使われます。電圧はVTGﾋﾟﾝ経由で印加され、さもなければ、2つの電力元が競合するでしょう。基本基板が接続される時にそれはVOFFﾋﾟﾝをlowに引きます。

VOFFﾋﾟﾝは何時でもLowに接続または開放することができ、この変更はNanoﾃﾞﾊﾞｯｶﾞへのﾋﾟﾝ変化割り込みによって検出され、これはその後にそれによって目的対象電圧調整器を制御します。

⚠警告 GNDへのVOFF短絡なしでのVTGﾋﾟﾝへの外部電圧印加は基板を恒久的な損傷にさせるかもしれません。

⚠警告 VOFFﾋﾟﾝへどんな電圧も印加しないでください。電源を許可するにはこのﾋﾟﾝを浮かせてください。

⚠警告 目的対象ﾃﾞﾊﾞｲｽの最高動作電圧より高い電圧の印加は基板を恒久的な損傷にさせるかもしれません。

情報: Nanoﾃﾞﾊﾞｯｶﾞが目的対象調整器を停止した場合、異常状態を示すために状態LEDの高速点滅を始めます。一旦異常が解決されると、Nanoﾃﾞﾊﾞｯｶﾞは目的対象調整器をONにして状態LEDの点滅を停止します。

書き込み、ﾃﾞﾊﾞｯｸﾞ、ﾃﾞｰﾀ流しは外部電源で未だ可能です。USBｹｰﾌﾞﾙはﾃﾞﾊﾞｯｶﾞと信号基準移転器に給電します。両調整器、ﾃﾞﾊﾞｯｶﾞ、基準移転器はUSBｹｰﾌﾞﾙが取り去られると電力断にされます。

情報: 目的対象ﾃﾞﾊﾞｲｽによって消費される電力に加えて、USBｹｰﾌﾞﾙが基板上のDEBUGｺﾈｸﾀに接続される時に基板上の基準移転器と電圧監視回路に給電するためにどの外部電源からも概ね100µAが引き出されます。USBｹｰﾌﾞﾙが接続されないと、基準移転器電圧ﾋﾟﾝに供給するために概ね5µAの最悪消費電流で多少の電流が使われます。代表的な値は100nA位の低さでしょう。

8.4. 電圧制御異常

本項は電源で起き得る最も一般的な問題を要約します。

目的対象電圧切断

目的対象部分が与えられた電圧で多すぎる電流を引き出す場合に設定した目的対象電圧に達しないことが起き得て、MIC5353調整器の過熱遮断安全機能を活性にさせます。これを避けるには目的対象部分の電流負荷を減らしてください。

目的対象電圧設定不到達

(4.4～5.25Vで指定される)USB入力電圧は与えられた電圧設定と消費電流に於いてMIC5353調整器の最大出力電圧を制限します。より高い出力電圧が必要とされる場合、より高い入力電圧のUSB電源を使うか、またはVTGﾋﾟﾝで外部電源を使ってください。

目的対象電圧が設定と違う

VOFFﾋﾟﾝをLowに設定することなくVTGﾋﾟﾝへ外部的に印加した電圧がこれを引き起こし得ます。目的対象電圧が電圧設定の上下100mVを超えて変動する場合、Nanoﾃﾞﾊﾞｯｶﾞがそれを検出し、内部電圧調整器が停止します。この問題を修正するにはVTGﾋﾟﾝから印加した電圧を取り去ってください。新しい状況が検出されると、Nanoﾃﾞﾊﾞｯｶﾞは基板上調整器を許可します。目的対象電圧が設定の100mV未満の場合にPS LEDが高速点滅しますが、設定の100mV越えよりも大きい時にONに留まることに注意してください。

非常に低い目的対象電圧またはなしでPS LEDが高速点滅

完全または部分的な短絡回路がこれを引き起こし得て、上で記述した問題の特別な場合です。短絡回路を取り除いてください。Nanoﾃﾞﾊﾞｯｶﾞが目的対象電圧調整器を再許可するでしょう。

目的対象電圧なしでPS LEDが点灯1

この状況は目的対象電圧が0.0Vに設定される場合に起きます。これを修正するには目的対象電圧を目的対象ﾃﾞﾊﾞｲｽに対して指定された電圧範囲内の値に設定してください。

目的対象電圧なしでPS LEDが点灯2

この状況は電力ｼﾞｬﾝﾊﾟを切断し、目的対象電圧調整器が目的対象ﾃﾞﾊﾞｲｽに対して指定された電圧範囲内の値に設定されている場合に起き得ます。これを修正するにはﾊﾟｯﾄﾞ間を線/橋渡しで半田付けするか、またはﾋﾟﾝ ﾍｯﾀﾞが実装されているならｼﾞｬﾝﾊﾟを追加してください。

VBUS出力電圧が低いまたは存在しない

VBUS出力電圧が低いまたは失っている場合、最も可能性の高い理由はVBUSでの高電流引き出しで、電流制限部を始動させ、VBUSを完全に遮断します。この問題を修正するにはVBUSﾋﾟﾝでの消費電流を減らしてください。

8.5. IDシステム

NanoﾃﾞﾊﾞｯｶﾞのID線はそれらの基本基板上のXplained Pro (XPRO)拡張ﾍｯﾀﾞに接続された基本基板と拡張を識別するのに使われる機構を支援します。

注: Microchip Xplained Pro拡張とCuriostiy Nano基本基板だけがこの識別ｼｽﾃﾑを特徴とします。

始動中、Nanoﾃﾞﾊﾞｯｶﾞは基本基板と拡張の両方に対してID線を走査し、要求に応じてこの情報をIDEへ提供します。

注: Nanoﾃﾞﾊﾞｯｶﾞ通電後に接続された拡張と基本基板は検出されません。

助言: IDｼｽﾃﾑのより多くの情報についてはXplained Proﾊｰﾄﾞｳｪｱ開発ｷｯﾄ(HDK)をご覧ください。

---

9. ﾃﾞﾊﾞｯｶﾞ構成設定

Nanoﾃﾞﾊﾞｯｶﾞは連携するﾊｰﾄﾞｳｪｱについての情報を格納するのに内部不揮発性ﾒﾓﾘの一部を使います。この内容は一般的に製造中に書かれませすが、最終使用者によって変更することもできます。この構成設定はﾃﾞﾊﾞｯｶﾞのﾌｧｰﾑｳｪｱが更新される時に保護され、従ってﾌｧｰﾑｳｪｱ自体で行われることから特別な独自化を保護するための重要な機構です。

ﾃﾞﾊﾞｯｶﾞ構成設定は次の2つの部分を含みます。

- 基板構成設定はNanoﾃﾞﾊﾞｯｶﾞが実装される基板を記述し、必須です。
- ﾃﾞﾊﾞｲｽ構成設定はｷｯﾄに恒久的に実装されるﾃﾞﾊﾞｲｽを記述し、任意選択です。

助言: ﾃﾞﾊﾞｲｽ構成設定が必要とされないなら、偽装構成設定が使われます。これはﾄﾞﾗｯｸﾞ&ﾄﾞﾛｯﾌﾟ書き込みが利用できないNanoﾃﾞﾊﾞｯｶﾞであることを示します。

9.1. 基板構成設定

Nanoﾃﾞﾊﾞｯｶﾞの基板構成設定(またはboad-config)は不揮発性ﾒﾓﾘの部分に属します。基板構成設定書式は各種の領域とﾋﾞｯﾄに割り当てられるこの部分での様々な変位(ｵﾌｾｯﾄ)と共にXMLで指定されます。ﾎｽﾄ ｺﾝﾋﾟｭｰﾀに対するAPIは書式の知識なしに単に塊読み書き操作です。領域の意味は構成設定XMLﾌｧｲﾙ自体から生成された(ﾎｽﾄ ｺﾝﾋﾟｭｰﾀに対する)Pythonﾌｧｲﾙと(Nanoﾃﾞﾊﾞｯｶﾞに対する)ﾍｯﾀﾞ ﾌｧｲﾙによって定義されます。新しい領域はXML内で未使用空間を割り当てることによってAPIを変更することなく追加することができ、Pythonとﾍｯﾀﾞのﾌｧｲﾙを生成します。XML構造は互換性のために版番号付け(SemVer)を使います。

表9-1. 基板構成設定 1.14

ﾚｼﾞｽﾀ	大きさ	説明	用途/既定値
DEVNAME	最大32 ASCII文字	装置名	ﾃﾞﾊﾞｲｽが基板に恒久的に実装される場合、ここに正確なﾃﾞﾊﾞｲｽ名を指定してください。 基板に目的対象ﾃﾞﾊﾞｲｽが実装されていない場合、この領域は空でなければなりません。
KITNAME	最大60 ASCII文字	ｷｯﾄ名	ここで素敵な名前を与えてください。
MNFRNAME	最大60 ASCII文字	製造業者名	供給者としてあなたを指定します。
SERNUM	20文字	USB通番	ここに特定のUSB通番を設定することができます。空白なしで正確に20個の大文字英数字を使うことが推奨されます。 “自動通番”ﾋﾞｯﾄを設定(1)すると、列挙時にこれを上書きしますが、元の値はﾌﾗｯｼｭ ﾒﾓﾘ内の構成設定に残ります。
REDIRECT	4文字	転送ID	下の注意をご覧ください。
DATE	8桁	製造日付	YYYYMMDD形式でｷｯﾄの製造日付を任意選択で指定します。
INSTANCE	1ﾊﾞｲﾄ	内容改訂	これは単純な版番号領域で、構成設定に変更が行われる毎に増すことができます。
TARGET_DEBUG_INTERFACES	4ﾊﾞｲﾄ	支援されるﾌﾟﾛｸﾞﾗﾑとﾃﾞﾊﾞｯｸﾞのｲﾝﾀｰﾌｪｽ一式	この実装でどの物理ｲﾝﾀｰﾌｪｰｽが支援されるかを示すためにこのﾋﾞｯﾄ領域のﾋﾞｯﾄを設定(1)。関連するﾋﾞｯﾄは、 ・ﾋﾞｯﾄ1: ARMﾃﾞﾊﾞｲｽ用SWD ・ﾋﾞｯﾄ8: AVRﾃﾞﾊﾞｲｽ用UPDI ・ﾋﾞｯﾄ10: ARMﾃﾞﾊﾞｲｽ用SWO ・ﾋﾞｯﾄ11: PICとdsPIC®ﾃﾞﾊﾞｲｽ用ICSP
TARGET_DEBUG_FEATURES	1ﾊﾞｲﾄ	実装されるﾌﾟﾛｸﾞﾗﾑとﾃﾞﾊﾞｯｸﾞの機能一式	個別機能を許可するためにこのﾋﾞｯﾄ領域のﾋﾞｯﾄを設定(1)。関連するﾋﾞｯﾄは、 ・ﾋﾞｯﾄ0: SINGLE_DEVICE – ﾃﾞﾊﾞｲｽが基板に恒久的に実装される場合にこのﾋﾞｯﾄを設定(1)。 ・ﾋﾞｯﾄ1: PROG_ENABLED – 書き込みを許可するにはこのﾋﾞｯﾄを設定(1)。 ・ﾋﾞｯﾄ2: DEBUG_ENABLED – ﾃﾞﾊﾞｯｸﾞを許可するにはこのﾋﾞｯﾄを設定(1)。

次頁へ続く

表9-1 (続き). 基板構成設定 1.14

ﾚｼﾞｽﾀ	大きさ	説明	用途/既定値
TARGET_DEBUG_FEATURES	1ﾊﾞｲﾄ	実装されるﾌﾟﾛｸﾞﾗﾑとﾃﾞﾊﾞｯｸﾞの機能一式	• ﾋﾞｯﾄ3: FUSE_CONFIG_PROTECTION – ﾋｭｰｽﾞと構成設定ﾊﾞｲﾄの保護を許可するにはこのﾋﾞｯﾄを設定(1)。これが働くには保護遮蔽一式がﾃﾞﾊﾞｲｽに提供されなければなりません。 • ﾋﾞｯﾄ4: AUTO_SERIAL_NUMBER – USB通番を自動的に生成するにはこのﾋﾞｯﾄを設定(1)
DGI_INTERFACES	4ﾊﾞｲﾄ	支援DGIｲﾝﾀｰﾌｪｰｽ一式	どのDGIｲﾝﾀｰﾌｪｰｽが接続されるかをﾃﾞﾊﾞｯｶﾞに示すためにこのﾋﾞｯﾄ領域のﾋﾞｯﾄを設定(1)。関連するﾋﾞｯﾄは、 • ﾋﾞｯﾄ0: GPIO
DGI_GPIO_MAP	1ﾊﾞｲﾄ	DGI GPIOﾁｬﾈﾙの配置	どのDGI GPIO線が接続されるかをﾃﾞﾊﾞｯｶﾞに示すためにこのﾋﾞｯﾄ領域のﾋﾞｯﾄを設定(1)。関連するﾋﾞｯﾄは、 • ﾋﾞｯﾄ0: DGI_GPIO0 • ﾋﾞｯﾄ1: DGI_GPIO1
ANALOG_FEATURES	4ﾊﾞｲﾄ	ｱﾅﾛｸﾞ機能	基板でどのｱﾅﾛｸﾞ実装が提供されるかをﾃﾞﾊﾞｯｶﾞに示すためにこのﾋﾞｯﾄ領域のﾋﾞｯﾄを設定(1)。関連するﾋﾞｯﾄは、 • ﾋﾞｯﾄ0: TVS – ﾃﾞﾊﾞｯｶﾞは目的対象電圧を制御することができます。 • ﾋﾞｯﾄ1: TVR – ﾃﾞﾊﾞｯｶﾞは目的対象電圧を測定することができます。 • ﾋﾞｯﾄ2: LVC – ﾃﾞﾊﾞｯｶﾞと目的対象間に基準移転器が存在します。
VMIN	1ﾊﾞｲﾄ	最小電圧	調節可能な電圧の基板に対し、0.1V単位で最小電圧を指定
VMAX	1ﾊﾞｲﾄ	最大電圧	調節可能な電圧の基板に対し、0.1V単位で最大電圧を指定
VTG	1ﾊﾞｲﾄ	既定電圧	調節可能な電圧の基板に対し、0.1V単位で既定電圧を指定
VREG	1ﾊﾞｲﾄ	電圧調整器	実装される電圧調整器を指定。値は、 • $00 – 調節不可 • $01 – Nanoﾃﾞﾊﾞｯｶﾞによって制御される調節可能調整器構成設定1
ID_CHANNELS	1ﾊﾞｲﾄ	IDﾁｬﾈﾙ	どのIDﾁｬﾈﾙが実装されるかを指定 • ﾋﾞｯﾄ0: IDﾁｬﾈﾙ1
CONFIG_FORMAT_MAJOR CONFIG_FORMAT_MINOR CONFIG_FORMAT_BUILD	4ﾊﾞｲﾄ	使われる構成設定仕様の版番号。新しい領域が追加された場合、版番号が更新され、この文書も更新されます。	(最新)
HARDWARE_MOD	1ﾊﾞｲﾄ	ﾌｧｰﾑｳｪｱに影響を及ぼすﾊｰﾄﾞｳｪｱ更新を指定	$00
USB_ENUMERATION	1ﾊﾞｲﾄ	USB列挙(接続認識)	$03
MSD_SETTINGS	1ﾊﾞｲﾄ	大容量記憶内容設定	$00
CLICK_ME_TYPE	1ﾊﾞｲﾄ	ここをｸﾘｯｸ型構成設定	$FF
I2C_ADDR_0, I2C_ADDR_1	1ﾊﾞｲﾄ	I²Cｱﾄﾞﾚｽ、代替I²Cｱﾄﾞﾚｽ	$00
LABS	1ﾊﾞｲﾄ	実験的ﾌｧｰﾑｳｪｱ機能有効	$00

次頁へ続く

表9-1 (続き). 基板構成設定 1.14

ﾚｼﾞｽﾀ	大きさ	説明	用途/既定値
KEY1_TYPE	1ﾊﾞｲﾄ	鍵形式: • 0: なし • 1: 48ﾋﾞｯﾄMAC • 2: 64ﾋﾞｯﾄMAC • 3: UUID • 4: UID • 5: PUBKEY	IoTｷｯﾄ
KEY2_TYPE	1ﾊﾞｲﾄ	鍵形式: • 0: なし • 1: 48ﾋﾞｯﾄMAC • 2: 64ﾋﾞｯﾄMAC • 4: UID	
KEY3_TYPE	1ﾊﾞｲﾄ		
KEY4_TYPE	1ﾊﾞｲﾄ		
KEY1_LEN	1ﾊﾞｲﾄ	最大64ﾊﾞｲﾄ	鍵1内容長
KEY2_LEN	1ﾊﾞｲﾄ	最大32ﾊﾞｲﾄ	鍵2内容長
KEY3_LEN	1ﾊﾞｲﾄ	最大16ﾊﾞｲﾄ	鍵3内容長
KEY4_LEN	1ﾊﾞｲﾄ	最大16ﾊﾞｲﾄ	鍵4内容長
KEY1	64ﾊﾞｲﾄ	鍵1値	–
KEY2	32ﾊﾞｲﾄ	鍵2値	–
KEY3	16ﾊﾞｲﾄ	鍵3値	–
KEY4	16ﾊﾞｲﾄ	鍵4値	–

注: REDIRECT_ID領域はUSB大容量記憶ﾃﾞｨｽｸでURLを自動入力するのに使われ、kits.microchip.com経由で製品固有のｳｪﾌﾞ頁へ転送します。あなたの製品に対してNaonoﾃﾞﾊﾞｯｶﾞの独自化を望む場合、Microchipｻﾎﾟｰﾄまたはedbg@microchip.comにお問い合わせください。

9.2. ﾃﾞﾊﾞｲｽ構成設定

Nanoﾃﾞﾊﾞｯｶﾞの装置構成設定はﾃﾞﾊﾞｯｶﾞに恒久的に接続されたﾃﾞﾊﾞｲｽについての情報を提供します。これの主な機能は(IDEが必要とされない)自己完結型を必要とするﾄﾞﾗｯｸﾞ&ﾄﾞﾛｯﾌﾟ書き込みを支援します。ﾃﾞﾊﾞｲｽ構成設定は空(非書き込み)にすることができますが、ﾄﾞﾗｯｸﾞ&ﾄﾞﾛｯﾌﾟ書き込みが支援されない場合(例えば、ARM目的対象ﾃﾞﾊﾞｲｽ)や、目的対象ﾃﾞﾊﾞｲｽが実装されない(例えば、独立型ﾃﾞﾊﾞｯｶﾞの計画の)時に無効(null)または偽装ﾃﾞﾊﾞｲｽ構成設定を書くことが推奨されます。

ﾃﾞﾊﾞｲｽ構成設定は単一の実体だけを持ち、使用者によって変更されることを意図されません。

Nanoﾃﾞﾊﾞｯｶﾞ ﾌｧｰﾑｳｪｱのzip内のｲﾒｰｼﾞは無効ﾃﾞﾊﾞｲｽ構成設定を含みます。

pypi.orgで入手可能なpydebuggerconfigﾂｰﾙはNanoﾃﾞﾊﾞｯｶﾞ上のﾃﾞﾊﾞｲｽ構成設定を書いたり置き換えたりするのに使うことができます。

ﾃﾞﾊﾞｲｽ構成設定　—高度な情報

重要: この情報は参照のみです。ﾃﾞﾊﾞｲｽ構成設定は使用者に提供されることを意図されていません。

ﾃﾞﾊﾞｲｽ構成設定はﾃﾞﾊﾞｯｶﾞ更新の間に保護されるNanoﾃﾞﾊﾞｯｶﾞの不揮発性ﾒﾓﾘ内の3Kﾊﾞｲﾄ領域です。

表9-2. 先頭部構造

変位	型	領域	説明
0	uint8	主版番号	ﾃﾞﾊﾞｲｽ特定構成設定仕様の主版番号
1	uint8	副版番号	副版番号
2	uint16	構築番号	仕様の構築番号
4	uint16	内容長	(先頭部の後の)実際の内容のﾊﾞｲﾄ数
6	uint16	内容ﾁｪｯｸｻﾑ	内容に渡るﾁｪｯｸｻﾑ (現在検証されていません。)
8	uint8	実体	この構成設定実体の版番号。順次増加
9	7ﾊﾞｲﾄ	(予約)	将来使うため

表9-3. 第1版符号化

変位	型	領域	説明
16	uint8 (enum)	書き込みｲﾝﾀｰﾌｪｰｽ形式	$00: なし $01: EDBG APIによるAVR UPDI $02: (予約) $03: GEN4ﾊﾞｲﾅﾘ ｺｰﾄﾞによるPIC $04: GEN4基本によるPIC $05～$FF: (予約)
17	uint8 (enum)	ﾃﾞﾊﾞｲｽ副変種	PICﾃﾞﾊﾞｲｽに対して、 ・$00: PIC16 ・$01: PIC18 ・$02: PIC24/dsPIC ・$03～$FF: (予約)
18	14ﾊﾞｲﾄ	(予約)	将来使うため

これらの先頭部の後で2進物(BLOB)の関連変位の参照表が符号化されます。

表9-4. BLOB表符号化

変位	型	領域	説明
32	uint8	'L'	一覧(List)通票
33	uint8	大きさ	項目数
34	uint16	変位	項目1の変位
～	～	～	～
34+2n	uint16	変位	項目nの変位

各BLOBが符号化されます。

表9-5. BLOB先頭部符号化

変位	型	領域	説明
0	uint8	構造体型	・'S': GEN4ｽｸﾘﾌﾟﾄからの2進物 ・'D': ﾃﾞﾊﾞｲｽ状況(AVR用に使われる)
1	uint8	ID	例: ｽｸﾘﾌﾟﾄID
2	uint16	大きさ	ﾊﾞｲﾄでのﾃﾞｰﾀ長
4～		ﾃﾞｰﾀ	2進ﾃﾞｰﾀ

表9-6. GEN4ｽｸﾘﾌﾟﾄ2進型

ID	機能
0	ﾌﾟﾛｸﾞﾗﾐﾝｸﾞ動作移行用2進符号
1	ﾃﾞﾊﾞｲｽID取得用2進符号
2	ﾁｯﾌﾟ消去用2進符号
3	ﾌﾗｯｼｭ ﾒﾓﾘ書き込み用2進符号
4	構成設定語書き込み用2進符号
5	ﾌﾟﾛｸﾞﾗﾐﾝｸﾞ動作抜け出し用2進符号
6	ﾌﾗｯｼｭ ﾒﾓﾘ読み込み用2進符号

ﾃﾞﾊﾞｲｽ状況 (要素によるﾃﾞﾊﾞｲｽ ﾃﾞｰﾀ)

AVRﾃﾞﾊﾞｲｽに対するﾃﾞﾊﾞｲｽ状況は組み込みﾃﾞﾊﾞｯｶﾞに基づくツール規約使用者の手引きとpymcuprogで記述されます。

表9-7. AVRﾃﾞﾊﾞｲｽ用ﾃﾞﾊﾞｲｽ状況

変位	型	説明
$00	uint16	ﾌﾗｯｼｭ ﾒﾓﾘ書き込みの基準ｱﾄﾞﾚｽ
$02	uint8	ﾌﾗｯｼｭ ﾒﾓﾘ ﾍﾟｰｼﾞのﾊﾞｲﾄ数
$03	uint8	EEPROMﾍﾟｰｼﾞのﾊﾞｲﾄ数
$04	uint16	NVMCTRL単位部のｱﾄﾞﾚｽ
$06	uint16	OCD単位部のｱﾄﾞﾚｽ
$08	uint16	(予約)
$0A	uint16	(予約)
$0C	uint16	(予約)
$0E	uint16	(予約)
$10	uint16	UPDIﾋﾟﾝの最大周波数
$12	uint32	ﾊﾞｲﾄでのﾌﾗｯｼｭ ﾒﾓﾘの大きさ
$16	uint16	ﾊﾞｲﾄでのEEPROMの大きさ
$18	uint16	ﾊﾞｲﾄでの使用者列の大きさ
$1A	uint8	ﾊﾞｲﾄでのﾋｭｰｽﾞ ﾒﾓﾘの大きさ
$1B	uint8	ﾋｭｰｽﾞ領域内のSYSCFG0またはPINCFGのﾋｭｰｽﾞの変位(ｵﾌｾｯﾄ)
$1C	uint8	書き込み時にSYSCFG0/PINCFG値に適用するAND遮蔽値
$1D	uint8	書き込み時にSYSCFG0/PINCFG値に適用するOR遮蔽値
$1E	uint8	消去後にSYSCFG0/PINCFG値に適用するAND遮蔽値
$1F	uint8	消去後にSYSCFG0/PINCFG値に適用するOR遮蔽値
$20	uint16	EEPROMﾒﾓﾘの基準ｱﾄﾞﾚｽ
$22	uint16	使用者列ﾒﾓﾘの基準ｱﾄﾞﾚｽ
$24	uint16	識票列ﾒﾓﾘの基準ｱﾄﾞﾚｽ
$26	uint16	ﾋｭｰｽﾞ ﾒﾓﾘの基準ｱﾄﾞﾚｽ
$28	uint16	施錠ﾋﾞｯﾄ ﾒﾓﾘの基準ｱﾄﾞﾚｽ
$2A	uint16	ﾃﾞﾊﾞｲｽID
$2C	uint8	ﾌﾟﾛｸﾞﾗﾑ用ﾌﾗｯｼｭ ﾒﾓﾘの基準ｱﾄﾞﾚｽのMSB (変位$00の拡張)
$2D	uint8	ﾊﾞｲﾄでのﾌﾗｯｼｭ ﾒﾓﾘ ﾍﾟｰｼﾞの大きさのMSB (変位$02の拡張)
$2E	uint8	UPDIのｱﾄﾞﾚｽ指定動作: • $00: 16ﾋﾞｯﾄ ｱﾄﾞﾚｽ指定を使用 • $01: 24ﾋﾞｯﾄ ｱﾄﾞﾚｽ指定を使用
$2F	uint8	高電圧実装
$30	uint16	ﾌﾞｰﾄ列の基準ｱﾄﾞﾚｽ
$32	uint16	ﾊﾞｲﾄでのﾌﾞｰﾄ列の大きさ
$34	uint16	ﾋｭｰｽﾞ保護遮蔽。(LSBから)ﾋｭｰｽﾞ ﾊﾞｲﾄ毎に1ﾋﾞｯﾄ: • 0: 書き込み許可 • 1: 保護

PICﾃﾞﾊﾞｲｽに対するﾃﾞﾊﾞｲｽ状況はここで記述されます。

表9-8. PICﾃﾞﾊﾞｲｽ用ﾃﾞﾊﾞｲｽ状況

変位	型	説明
$00	uint32	PFM(ﾌﾟﾛｸﾞﾗﾑ用ﾌﾗｯｼｭ ﾒﾓﾘ)の基準ｱﾄﾞﾚｽ
$04	uint32	EEPROMﾒﾓﾘの基準ｱﾄﾞﾚｽ
$08	uint32	使用者列ﾒﾓﾘの基準ｱﾄﾞﾚｽ
$0C	uint32	構成設定ﾒﾓﾘの基準ｱﾄﾞﾚｽ
$10	uint32	ﾊﾞｲﾄでのﾌﾗｯｼｭ ﾒﾓﾘの大きさ
$14	uint16	ﾊﾞｲﾄでのEEPROMの大きさ
$16	uint16	ﾊﾞｲﾄでの使用者列の大きさ
$18	uint16	ﾊﾞｲﾄでの構成設定ﾒﾓﾘの大きさ
$1A	uint16	ﾊﾞｲﾄでのﾌﾗｯｼｭ ﾒﾓﾘ書き込み塊の大きさ
$1C	uint16	ﾊﾞｲﾄでのEEPROM書き込み塊の大きさ
$1E	uint8	ﾊﾞｲﾄでの使用者列書き込み塊の大きさ
$1F	uint8	ﾊﾞｲﾄでの構成設定ﾒﾓﾘ書き込み塊の大きさ
$20	uint16	ﾃﾞﾊﾞｲｽID
$22	uint32	構成設定ﾒﾓﾘ保護遮蔽。(LSBから)構成設定位置毎に1ﾋﾞｯﾄ: • 0: 書き込み許可 • 1: 保護

9.3. ﾃﾞﾊﾞｯｶﾞ構成設定の変更

基板構成設定とﾃﾞﾊﾞｲｽ構成設定はpydebuggerconfigﾕｰﾃｨﾘﾃｨを使って最終使用者によって変更することができます。

pydebuggerconfig使用についてのより多くの一般的な情報に関してはpypi.orgでの情報をご覧ください。

使用事例: ﾃﾞﾊﾞｲｽ保護用構成設定変更

ｷｯﾄに恒久的に実装されたAVRﾃﾞﾊﾞｲｽと共に使われる時に、Nanoﾃﾞﾊﾞｯｶﾞは特にﾄﾞﾗｯｸﾞ&ﾄﾞﾛｯﾌﾟ書き込み使用時、回復不能になることからMCUを守ることを意図される保護機構を含みます。書き込みとﾃﾞﾊﾞｯｸﾞのｲﾝﾀｰﾌｪｰｽの禁止や定常的なﾒﾓﾘ施錠は開発ｷｯﾄ基盤での評価に貢献しないいくつかのMCUの機能です。

この保護機構は関連するﾋｭｰｽﾞや構成設定のﾋﾞｯﾄに対する書き込み操作を阻止することによって働き、書かれるｱﾄﾞﾚｽと値を条件付きで遮蔽します。

情報: 保護機構は予期せず元に戻せない変更を行うことを防ぐのを意図されます。使用者が元に戻せない変更を行うには自己責任で行ってください。

情報: 利用可能な機能一式と対応する保護機構はﾃﾞﾊﾞｲｽに依存します。更なる情報についてはﾃﾞｰﾀｼｰﾄをご覧ください。

元に戻せない変更を必要とする機能の完全な体験を望む使用者に関しては保護機構を禁止することができます。それを行うと、恒久的な変更になります。

pydebuggercofigは構成設定保護を含むﾃﾞﾊﾞｯｶﾞの多くの面を微調整するのに使うことができます。この操作は最新公開のPython 3とその環境でのpydebuggercofig一括のｲﾝｽﾄｰﾙを必要とします。

手順1: 現在の状態

次を実行することによって保護が現在許可されているかどうかを判定してください。

```
pydebuggerconfig read
```

次に、以下の部分を調べてください。

```
Register TARGET_DEBUG_FEATURES:   0x0F (15)   # ﾌﾟﾛｸﾞﾗﾑ/ﾃﾞﾊﾞｯｸﾞ機能
            bit 0, SINGLE_DEVICE: 1 # 単一ﾃﾞﾊﾞｲｽ
            bit 1, PROG_ENABLED: 1 # ﾌﾟﾛｸﾞﾗﾐﾝｸﾞ
            bit 2, DEBUG_ENABLED: 1 # ﾃﾞﾊﾞｯｸﾞ
            bit 3, FUSE_PROTECTION: 1 # ﾋｭｰｽﾞ保護
```

‘1’のFUSE_PROTECTION値は保護が実施されていることを示します。

手順2: 保護設定変更

FUSE_PROTECTIONを'0'に設定することによってTARGET_DEBUG_FEATURESに対して望む設定を計算してください。例えば次を実行することによってTARGET_DEBUG_FEATURESﾚｼﾞｽﾀの現在の値を置き換えてください。

```
pydebuggerconfig replace -r TARGET_DEBUG_FEATURES=0x07
```

新しい値を確認するため、**手順1**の手続きを繰り返し、その後にｷｯﾄの電源をOFF/ONしてください。全ての保護機構が今や禁止されています。

手順3: 保護を復元

保護は初期値で**手順2**を繰り返すか、または次を実行することによって工場復元機能を使うかのどちらかで再許可することができます。

```
pydebuggerconfig restore
```

重要: 復元機能はﾃﾞﾊﾞｯｶﾞで内部的に格納されたｷｯﾄ構成設定の複製を復元することにより、製造から変更されたどのｷｯﾄ構成設定も復元するだけです。この操作はMCUに影響を及ぼさず、その構成設定に行われた元に戻せないどの変更も元に戻しません。

10. ツールとIDE

Nanoﾃﾞﾊﾞｯｶﾞ はMicrochip、我々の協力企業、開放ソース共同体によって公開された多数の前置ツールで支援されます。

10.1. 協力社収益活動協調体制

Nanoﾃﾞﾊﾞｯｶﾞ は様々な協力社収益活動協調体制とIDEによって支援されます。CMSIS-DAP標準ﾃﾞﾊﾞｯｸﾞ実装に基づき、どのARM Cortexﾃﾞﾊﾞｲｽに対しても効率的な支援を提供します。AVR規約仕様は公開で、AVRﾃﾞﾊﾞｲｽを支援する様々なIDEがNanoﾃﾞﾊﾞｯｶﾞ も支援します(これはJTAGICE3、Atmel-ICE、Powerﾃﾞﾊﾞｯｶﾞ、EDBGで使われるのと同じ規約です)。

協力社統合は以下を含みますが、これに限定されません。

- IAR Embedded Workbench
- Keil Microvision
- CodeVisionAVR
- OpenOCD
- pyOCD
- AVRDUDE
- PlatformIO
- Bloom
- probe-rs

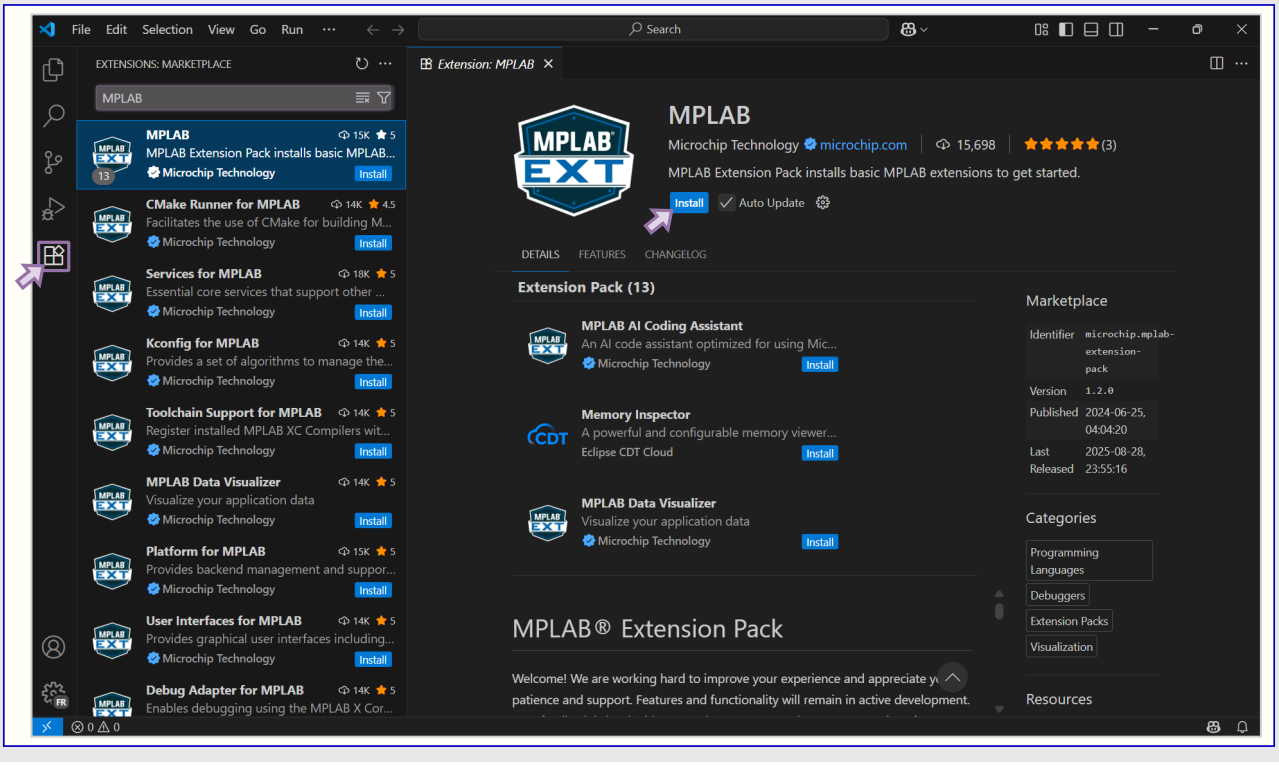
助言: 我々の協力社についての最新情報に関しては開発ツール協力社をご覧ください。

10.2. VS Code用MPLAB®ツール

Nanoﾃﾞﾊﾞｯｶﾞ はVS Code開発環境で支援されます。標準CMSIS-DAPﾃﾞﾊﾞｯｶﾞ として、一般的なARM Cortexﾃﾞﾊﾞｲｽ用ﾃﾞﾊﾞｯｸﾞ ｱﾀﾞﾌﾟﾀとそれらの機構を使って構築された枠組みを支援します。加えて、VS Code用MPLABツールによって支援されます。

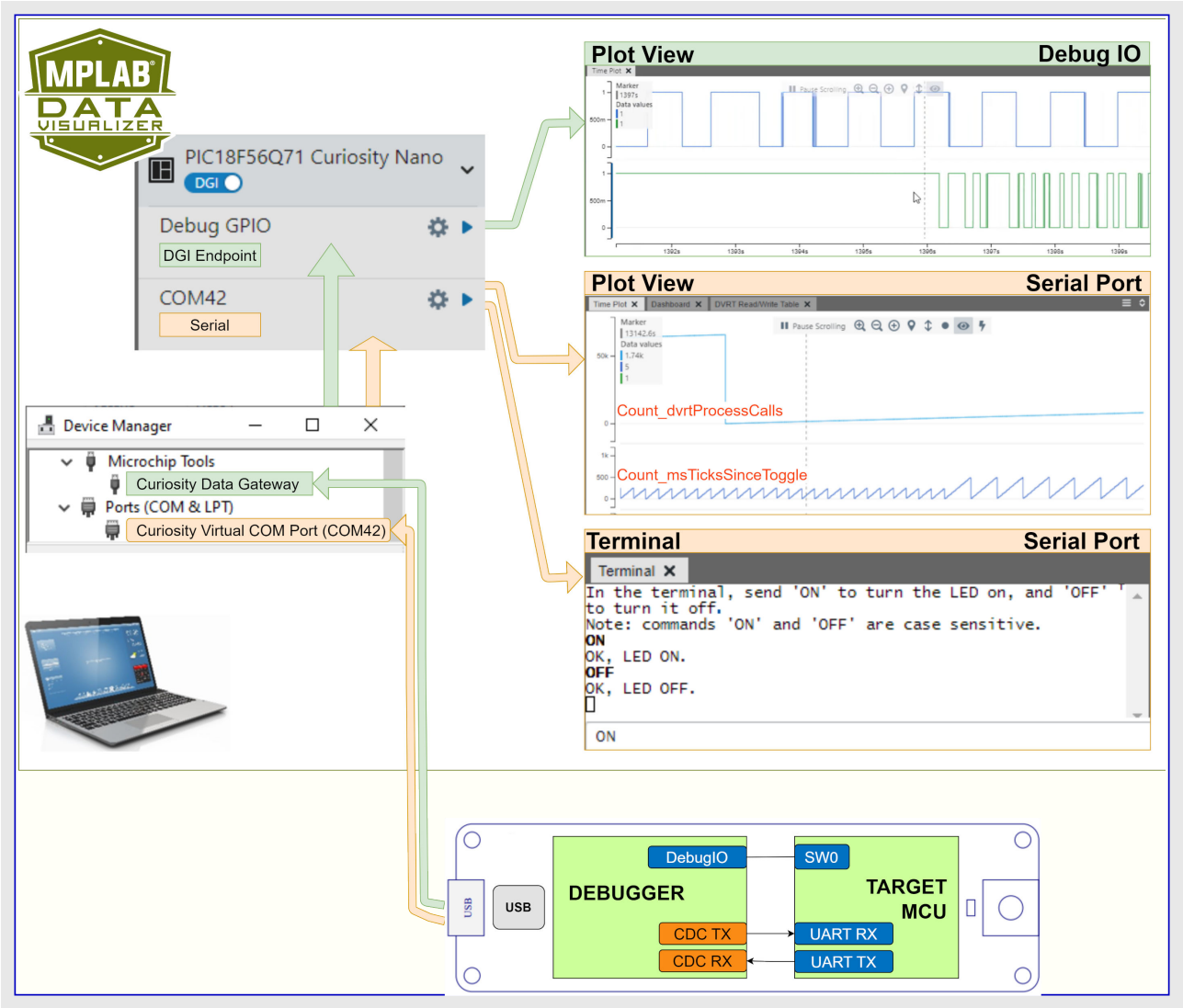
VS Code用MPLABツールはMPLAB拡張公式ｳｪﾌﾞｻｲﾄまたはVS Code市場からﾀﾞｳﾝﾛｰﾄﾞすることができます。各資源は個別にまたはMPLAB拡張一括の一部として共にｲﾝｽﾄｰﾙすることができるMPLAB拡張の完全な一覧を提供します。ｲﾝｽﾄｰﾙはVS Code拡張を通して実行されます。指導書は拡張開発者ﾍﾙﾌﾟとVS Code映像再生一覧用MPLAB拡張へのﾘﾝｸと共に公式ｳｪﾌﾞｻｲﾄで入手可能です。

図10-1. VS Code拡張市場でのMPLAB拡張



10.3. MPLABデータ可視器の使用

Nanoデバッガはそれの USB/シリアル橋渡し経由で目的対象MCU上のUARTとコンピュータのCOMポート間の接続を支援します。例えば、MPLAB Data Visualizer(データ可視器)や他の端末プログラムへ接続するのにこれを使うことができます。



助言: 上のデータ可視器図と端末画面の例に対する参照

1. Plot View – DebugIO: DebugIO Hello World (Microchip大学)
2. Plot View – Serial Port: MCC Melody使用事例、データ可視器走行時使用事例1
3. Terminal – Serial Port: MCC Melody UARTドライバ: LED制御指令

10.4. キット ウィンドウ表示部

MicrochipのIDEはUSB経由で接続されたMicrochipツールを検出するための機構を持ち、キット ウィンドウで有用な情報を提供します。これには例えば、当該のキットの画像、キットやそれに実装されるデバイスの注文情報、回路図、例プロジェクトへのリンクを含みます。

この表示部は各IDEで異なる外観と操作感で表現されますが、そのデータはウェブ サーバーに配置されます。データベースの更新はどのIDE公開時期とも同期されません。

キット ウィンドウで提供されるデータのいくつかは接続したキットから読まれ、データベースから取得されません。このデータはNanoデバッガの基板構成設定に格納され、従ってキット開発者(協力社)だけでなく最終使用者によっても変更され得ます。

注: このキット ウィンドウはMicrochipによって開発された機構です。このキット情報の一部は協力社のIDEとツールで標準CMSIS-DAP API命令を使ってアクセス可能です。

キット ウィンドウ表示部の例についてはMPLAB Xをご覧ください。

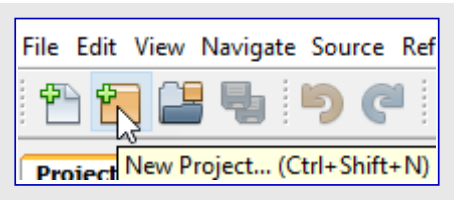
10.5. MPLAB X

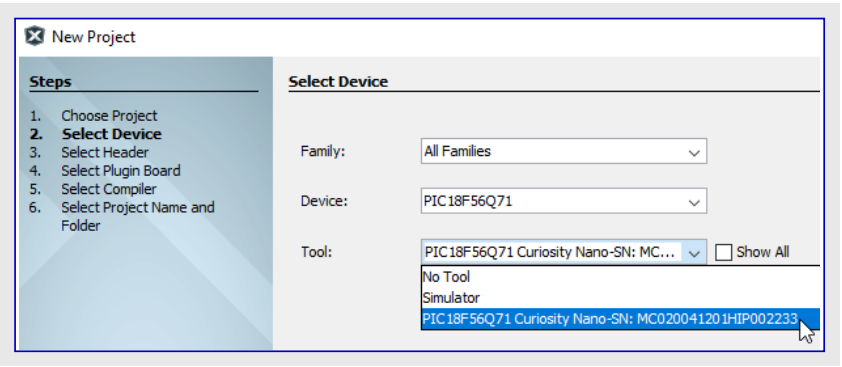
NanoﾃﾞﾊﾞｯｶﾞはMPLAB X とIPEで支援されます。

助言: MicrochipはMPLAB X使用者に対して有益な更新であるVS Code用MPLABツールを公開しました!。

NanoﾃﾞﾊﾞｯｶﾞがUSB経由でﾎｽﾄPCに接続される時にMPLAB X IDEが開くと、提供された鍵となるいくつかのﾘﾝｸと共にｷｯﾄ ｳｨﾝﾄﾞｳが開かれます。

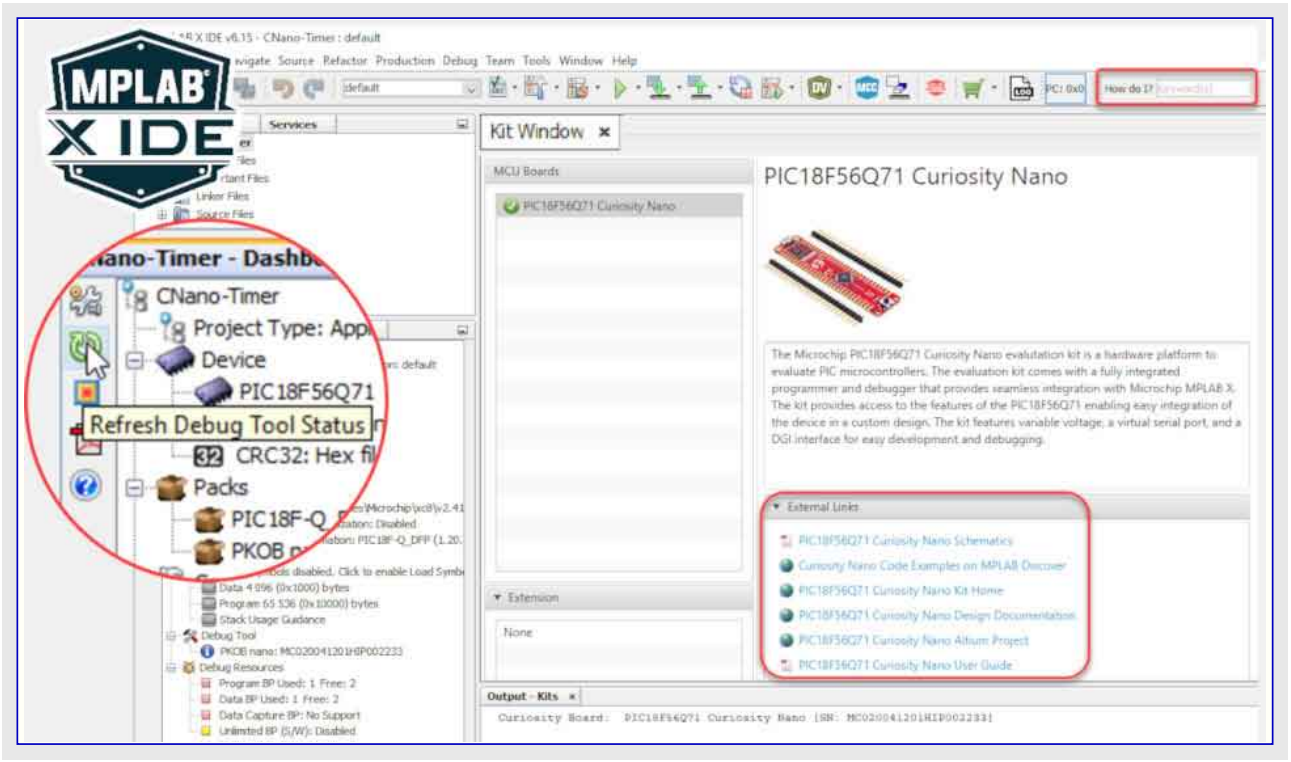
Nanoﾃﾞﾊﾞｯｶﾞを含む開発基板(例えば、Curiosity Nanoｷｯﾄ)を使って新規ﾌﾟﾛｼﾞｪｸﾄを作成すると、接続した基板から部品番号が読まれます。独立型構成設定のNanoﾃﾞﾊﾞｯｶﾞはこの情報を持たず、故に部品番号は手動で入力されなければなりません。



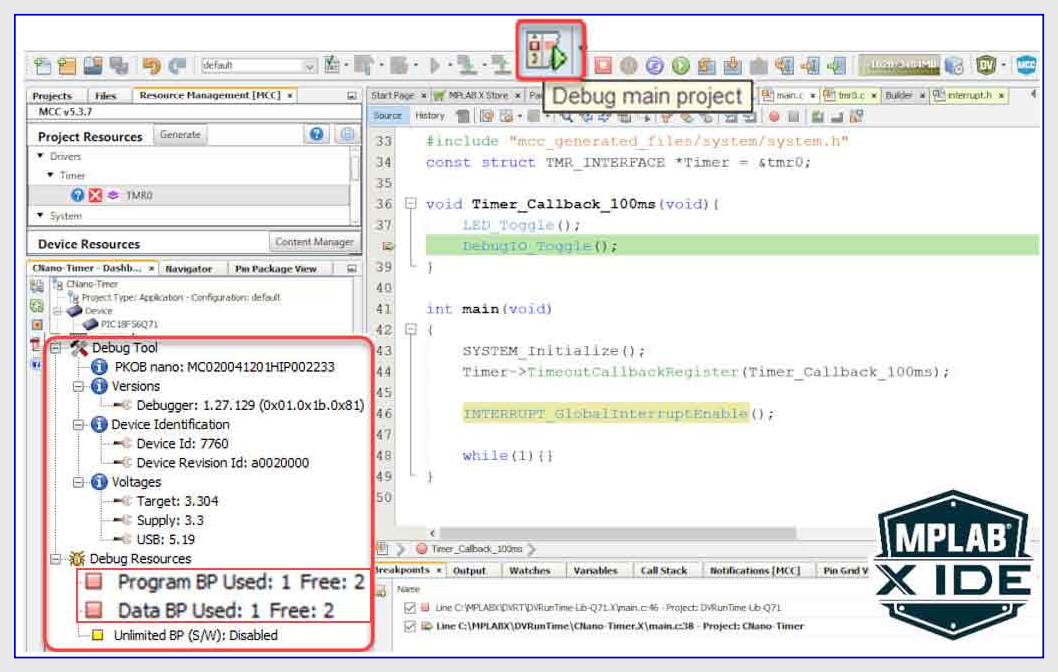


注: 特に新規ﾌﾟﾛｼﾞｪｸﾄを開始する時に最新のﾂｰﾙ一括とﾃﾞﾊﾞｲｽ系統一括を使うことを確実にしてください。

下図で示されるように、一旦"Refresh Debug Tool Status(ﾃﾞﾊﾞｯｸﾞ ﾂｰﾙ状態更新)"をｸﾘｯｸすると、そのCuriosity Nanoについての追加情報を"Debug Tool(ﾃﾞﾊﾞｯｸﾞ ﾂｰﾙ)"ｳｨﾝﾄﾞｳで見ることができます。



助言: • 閉じている場合、メニューバーでWindow(ウィンドウ)⇒Kit Window(キット ウィンドウ)を通して再びMPLAB® X IDEでキット ウィンドウを開くことができます。
• MPLAB X IDEが初めての場合、"How do I?(どうしますか?)"検索バーは度々素晴らしい結果を提供します。
• "Debug main project(主プロジェクトをデバッグ)"はデバッグ作業を開始します。



助言: • "Refresh Debug Tool Status(デバッグ ツール状態更新)"をクリック後、MCU目的電圧のような情報を見ることができます。
• プログラムとデータの中断点(ブレーク ポイント)とは何ですか?。MPLAB X IDE高度なデバッグ – 中断点実演

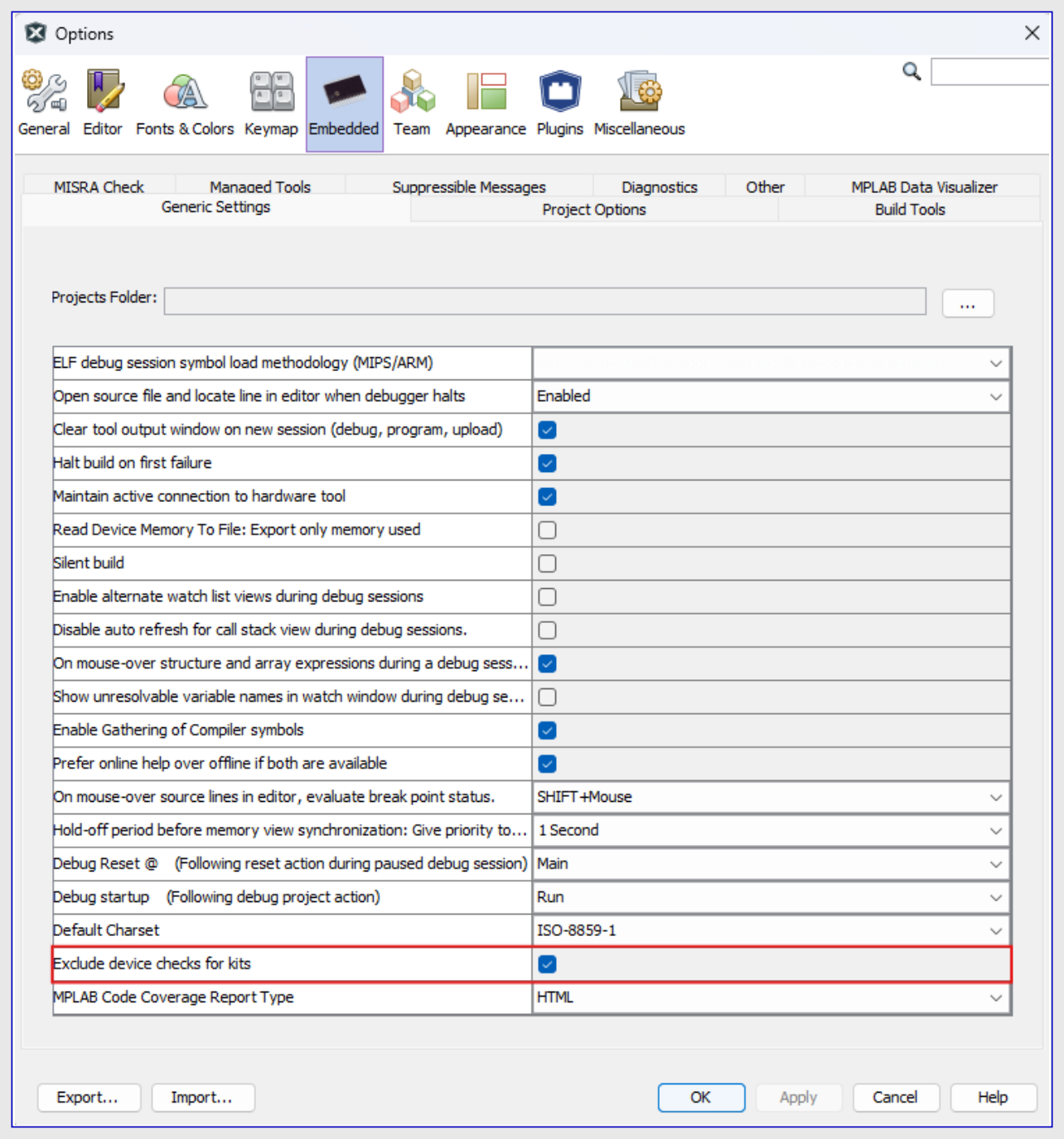
外部マイクロ コントローラとでのNanoデバッガ キットの使用

開発キット上のNanoデバッガは元々そのキットに実装されていないマイクロ コントローラとで動作するように簡単に変更することができます。

注: Curiosity Nanoキットとでこの手順を実行する時にその特定キット用の使用者の手引きをご覧ください。

基板上に実装されたものと異なるマイクロ コントローラを書いてデバッグするにはデバイスと書き込みインターフェースの独立した選択を許すようにMicrochip MPLAB X IDEを構成設定してください。

1. 応用の上部のメニューを通してTools(ツール)⇒Options(任意選択)と誘導してください。
2. 任意選択ウィンドウでEmbedded(組み込み)⇒Generic Settings(全般選択)区分を選んでください。
3. Exclude device checks for kits(キット用デバイス検査を除外)任意選択をチェックしてください。



注: Microchip MPLAB X IDEは**Exclude device checks for kits**(ｷｯﾄ用ﾃﾞﾊﾞｲｽ検査を除外)設定が**ﾁｪｯｸされた**時にどのﾏｲｸﾛ ｺﾝﾄﾛｰﾗとｲﾝﾀｰﾌｪｰｽも許し、また、Nanoﾃﾞﾊﾞｯｶﾞによって支援されないﾏｲｸﾛ ｺﾝﾄﾛｰﾗとｲﾝﾀｰﾌｪｰｽも許します。

10.6. Microchip Studio

NanoﾃﾞﾊﾞｯｶﾞはMicrochip Studioによって支援され、一括ｻｰﾊﾞｰを通して配布されたﾃﾞﾊﾞｲｽとで使う時にMicrochip Studioで支援されます。これはいくつかのSAMﾃﾞﾊﾞｲｽとUPDIｲﾝﾀｰﾌｪｰｽを持つ多くのAVRﾃﾞﾊﾞｲｽを含みます。

Microchip Studioは”更新終了”で新規設計には推奨されません。

重要: NanoﾃﾞﾊﾞｯｶﾞUSBｲﾝﾀｰﾌｪｰｽは将来変更するかもしれず、Microchip Studioで使えないﾌｧｰﾑｳｪｱを提供し得ます。

助言: MicrochipはVS Code用MPLABﾂｰﾙを公開し、これはMicrochip Studio使用者にとって有益な更新です。

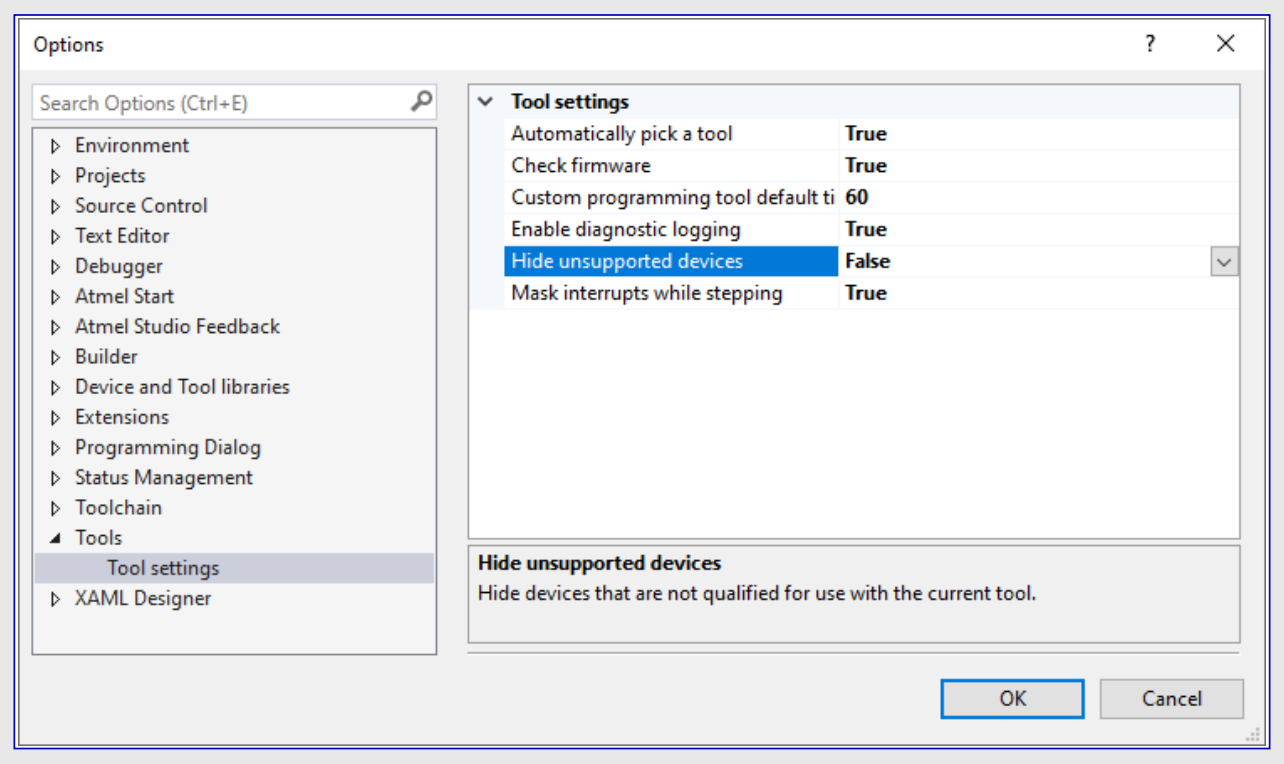
外部マイクロ コントローラとでのNanoﾃﾞﾊﾞｯｶﾞ ｷｯﾄの使用

開発ｷｯﾄ上のNanoﾃﾞﾊﾞｯｶﾞは元々そのｷｯﾄに実装されていないﾏｲｸﾛ ｺﾝﾄﾛｰﾗとで動作するように簡単に変更することができます。

注: Curiosity Nanoｷｯﾄとでこの手順を実行する時にその特定ｷｯﾄ用の使用者の手引きをご覧ください。

基板上に実装されたものと異なるﾏｲｸﾛ ｺﾝﾄﾛｰﾗを書いてﾃﾞﾊﾞｯｸﾞするにはﾃﾞﾊﾞｲｽと書き込みｲﾝﾀｰﾌｪｰｽの独立した選択を許すようにMicrochip Studioを構成設定してください。

1. 応用の上部のﾒﾆｭｰを通してTools(ﾂｰﾙ)⇒Options(任意選択)と誘導してください。
2. 任意選択ｳｨﾝﾄﾞｳでTools(ﾂｰﾙ)⇒Tool Settings(ﾂｰﾙ選択)区分を選んでください。
3. Hide unsupported devices(未支援ﾃﾞﾊﾞｲｽを隠す)任意選択をFalse(偽)にしてください。



注: Microchip StudioはHide unsupported devices(未支援ﾃﾞﾊﾞｲｽを隠す)設定がFalse(偽)に設定された時にどのﾏｲｸﾛ ｺﾝﾄﾛｰﾗとｲﾝﾀｰﾌｪｰｽも選ばれることを許し、また、基板上ﾃﾞﾊﾞｯｶﾞによって支援されないﾏｲｸﾛ ｺﾝﾄﾛｰﾗとｲﾝﾀｰﾌｪｰｽも許します。

10.7. Nanoﾃﾞﾊﾞｯｶﾞ ｷｯﾄと他のﾊｰﾄﾞｳｪｱ ﾂｰﾙ使用

開発基板の基板上ﾃﾞﾊﾞｯｶﾞとして実装された時にNanoﾃﾞﾊﾞｯｶﾞは外部ﾂｰﾙの接続を許します。そのようなﾂｰﾙは追加の機能やより良い性能を提供し得ます。

NanoﾃﾞﾊﾞｯｶﾞはIDEからの指令なしにﾃﾞﾊﾞｯｶﾞの信号線を駆動しません。

助言: Nanoﾃﾞﾊﾞｯｶﾞ ｷｯﾄと共に外部ﾃﾞﾊﾞｯｶﾞを使う情報についてはそのｷｯﾄの使用者の手引きを読んでください。

重要: 高電圧書き込みの能力を持つ書き込み器使用時、高電圧線からNanoﾃﾞﾊﾞｯｶﾞを切断することを確実にしてください。高電圧はNanoﾃﾞﾊﾞｯｶﾞに恒久的な損傷を与え得ます。

10.8. USBﾄﾞﾗｲﾊﾞ

Nanoﾃﾞﾊﾞｯｶﾞを持つ基板がｺﾝﾋﾟｭｰﾀへ最初に接続される時にｵﾍﾟﾚｰﾃｨﾝｸﾞ ｼｽﾃﾑはﾄﾞﾗｲﾊﾞ ｿﾌﾄｳｪｱをｲﾝｽﾄｰﾙします。Nanoﾃﾞﾊﾞｯｶﾞ用のﾄﾞﾗｲﾊﾞはVS Code用MPLABﾂｰﾙ、MPLAB X IDE、Microchip Studioに含まれています。

11. Pythonツール

Nanoﾃﾞﾊﾞｯｶﾞと共に動く時に有用なPythonに基づくいくつかのユーティリティがpypi.orgで公開されています。

11.1. pydebuggerupgrade

pydebuggerupgradeはCLIでNanoﾃﾞﾊﾞｯｶﾞ上のファームウェアを更新するためのﾗｲﾌﾞﾗﾘです。

何故それを使うのか

Nanoﾃﾞﾊﾞｯｶﾞのファームウェアを最新に保つのにpydebuggerupgradeを使ってください。

例

```
$ pydebuggerupgrade latest
Upgrading nedbg (ATML3203081800012252) to 'latest'
Upgrade to firmware version '1.33.76' successful
```

表11-1. 要約

性質	値
ﾀﾞｳﾝﾛｰﾄﾞ ﾘﾝｸ	pypi.org/project/pydebuggerupgrade/
ｿｰｽ ｺｰﾄﾞ	–
文書資料	–
ｺﾏﾝﾄﾞ行ｲﾝﾀｰﾌｪｰｽ	はい
ﾗｲﾌﾞﾗﾘ	はい
支援するｷｯﾄ形式	Nanoﾃﾞﾊﾞｯｶﾞに基づく
支援するﾃﾞﾊﾞｲｽ形式	全て

11.2. pykitinfo

pykitinfoはNanoﾃﾞﾊﾞｯｶﾞを含む接続した開発ｷｯﾄとﾃﾞﾊﾞｯｶﾞに関連する有用な情報を読み出すための軽量なPythonﾕｰﾃｨﾘﾃｨです。

何故それを使うのか

pykitinfo CLIはｺﾝﾋﾟｭｰﾀに接続した複数のNanoﾃﾞﾊﾞｯｶﾞ ｷｯﾄを持ち、どのｷｯﾄがどの仮想ｼﾘｱﾙ ﾎﾟｰﾄに割り当てられているかを素早く見る時に特に有用です。

例

```
$ pykitinfo
Looking for Microchip kits...
Compatible kits detected: 1
Kit ATML3203081800012252: 'AVR-IoT WG' (ATmega4808) on COM163
```

表11-2. 要約

性質	値
ﾀﾞｳﾝﾛｰﾄﾞ ﾘﾝｸ	pypi.org/project/pykitinfo/
ｿｰｽ ｺｰﾄﾞ	github.com/microchip-pic-avr-tools/pykitinfo
文書資料	microchip-pic-avr-tools.github.io/pykitinfo/
ｺﾏﾝﾄﾞ行ｲﾝﾀｰﾌｪｰｽ	はい
ﾗｲﾌﾞﾗﾘ	はい
支援するｷｯﾄ形式	様々
支援するﾃﾞﾊﾞｲｽ形式	全て

11.3. pyedbglib

pyedbglibはNanoﾃﾞﾊﾞｯｶﾞのCMSIS-DAPに組み込まれた規約と副規約へのｱｸｾｽを提供するﾗｲﾌﾞﾗﾘです。

何故それを使うのか

pyedbglibは独自のﾂｰﾙやﾕｰﾃｨﾘﾃｨを構築する時に有用で有り得ます。

例 (Pythonﾗｲﾌﾞﾗﾘ)

```
"""
Example usage of pyedbglib to read debugger firmware version and target voltage
"""
from pyedbglib.hidtransport.hidtransportfactory import hid_transport
from pyedbglib.protocols.housekeepingprotocol import Jtagice3HousekeepingProtocol
from pyedbglib import __version__ as pyedbglib_version

# Report library version
print("pyedbglib version {}".format(pyedbglib_version))

# Make a connection using HID transport
transport = hid_transport()
transport.connect()

# Create a housekeeper
housekeeper = Jtagice3HousekeepingProtocol(transport)
housekeeper.start_session()

# Read out debugger firmware version
major = housekeeper.get_byte(Jtagice3HousekeepingProtocol.HOUSEKEEPING_CONTEXT_CONFIG,
                             Jtagice3HousekeepingProtocol.HOUSEKEEPING_CONFIG_FWREV_MAJ)
minor = housekeeper.get_byte(Jtagice3HousekeepingProtocol.HOUSEKEEPING_CONTEXT_CONFIG,
                             Jtagice3HousekeepingProtocol.HOUSEKEEPING_CONFIG_FWREV_MIN)
build = housekeeper.get_le16(Jtagice3HousekeepingProtocol.HOUSEKEEPING_CONTEXT_CONFIG,
                             Jtagice3HousekeepingProtocol.HOUSEKEEPING_CONFIG_BUILD)
print ("Debugger firmware is version {}.{}.{}".format(major, minor,build))

# Read out target voltage
target_voltage =
housekeeper.get_le16(Jtagice3HousekeepingProtocol.HOUSEKEEPING_CONTEXT_ANALOG,
                                    Jtagice3HousekeepingProtocol.HOUSEKEEPING_ANALOG_VTREF)
print ("Target voltage is {:.02f}V".format(target_voltage/1000.0))

# Tear down
housekeeper.end_session()
transport.disconnect()
```

表11-3. 要約

性質	値
ﾀﾞｳﾝﾛｰﾄﾞ ﾘﾝｸ	pypi.org/project/pyedbglib/
ｿｰｽ ｺｰﾄﾞ	github.com/microchip-pic-avr-tools/pyedbglib
文書資料	microchip-pic-avr-tools.github.io/pyedbglib/
ｺﾏﾝﾄﾞ行ｲﾝﾀｰﾌｪｰｽ	はい
ﾗｲﾌﾞﾗﾘ	はい
支援するｷｯﾄ形式	様々
支援するﾃﾞﾊﾞｲｽ形式	全て

11.4. pymcuprog

pymcuprogはNanoﾃﾞﾊﾞｯｶﾞと他のMicrochip独立型ﾃﾞﾊﾞｯｶﾞに接続されたﾃﾞﾊﾞｲｽ上の不揮発性ﾒﾓﾘの消去、読み込み、書き込み用の軽量Pythonﾕｰﾃｨﾘﾃｨです。

何故それを使うのか

pymcuprog CLIはｺﾏﾝﾄﾞ行からのﾃﾞﾊﾞｲｽに対する接続性検査(”ping”)や簡単な読み書き操作を望む時に特に有用です。

例

```
$ pymcuprog ping
Connecting to anything possible
Connected to nEDBG CMSIS-DAP from Microchip (serial number ATML3203081800012252)
Debugger firmware version 1.21.37
Debugger hardware revision 0
Device mounted: 'atmega4808'
No device specified.  Using on-board target (atmega4808)
Pinging device...
Ping response: 1E9650
Done.
```

表11-4. 要約

性質	値
ﾀﾞｳﾝﾛｰﾄﾞ ﾘﾝｸ	pypi.org/project/pymcuprog/
ｿｰｽ ｺｰﾄﾞ	github.com/microchip-pic-avr-tools/pymcuprog
文書資料	microchip-pic-avr-tools.github.io/pymcuprog/
ｺﾏﾝﾄﾞ行ｲﾝﾀｰﾌｪｰｽ	はい
ﾗｲﾌﾞﾗﾘ	はい
支援するｷｯﾄ形式	• Curiosity Nano • 独立型CMSIS-DAP v1ﾃﾞﾊﾞｯｶﾞ
支援するﾃﾞﾊﾞｲｽ形式	• AVRﾃﾞﾊﾞｲｽ • Curiosity Nano基板を持つPICﾃﾞﾊﾞｲｽ • 選ばれたSAMDﾃﾞﾊﾞｲｽの限定的支援

11.5. pydebuggerconfig

pydebuggerconfigはNanoﾃﾞﾊﾞｯｶﾞを構成設定するためのﾕｰﾃｨﾘﾃｨです。

何故それを使うのか

Nanoﾃﾞﾊﾞｯｶﾞと共に独自のﾊｰﾄﾞｳｪｱを作成する時にpydebuggerconfigを使ってください。

例

```
$ pydebuggerconfig write -b board-configs/ATmega4809-CNANO.xml
```

表11-5. 要約

性質	値
ﾀﾞｳﾝﾛｰﾄﾞ ﾘﾝｸ	pypi.org/project/pydebuggerconfig/
ｿｰｽ ｺｰﾄﾞ	–
文書資料	–
ｺﾏﾝﾄﾞ行ｲﾝﾀｰﾌｪｰｽ	はい
ﾗｲﾌﾞﾗﾘ	はい
支援するｷｯﾄ形式	Nanoﾃﾞﾊﾞｯｶﾞ
支援するﾃﾞﾊﾞｲｽ形式	全て

11.6. pycmsisdapswitcher

pycmsisdapswitcherは或るMicrochipﾃﾞﾊﾞｯｶﾞとｷｯﾄ上のﾌｧｰﾑｳｪｱを切り替えるためのﾕｰﾃｨﾘﾃｨです。

重要: pycmsisdapswitcherはNanoﾃﾞﾊﾞｯｶﾞでは動かず、pydebuggerupgradeをご覧ください。

何故それを使うのか

PICkit™ 4、PICkit™ 5、PICkit™ Basic、Snap、PKOB41をCMSIS-DAP V"動作へ切り換えるにはpycmsisdapswitcherを使ってください。

例

```
$ pycmsisdapswitcher  --appsource path_to_appfile
```

表11-6. 要約

性質	値
ﾀﾞｳﾝﾛｰﾄﾞ ﾘﾝｸ	pypi.org/project/pycmsisdapswitcher/
ｿｰｽ ｺｰﾄﾞ	–
文書資料	–
ｺﾏﾝﾄﾞ行ｲﾝﾀｰﾌｪｰｽ	はい
ﾗｲﾌﾞﾗﾘ	はい
支援するｷｯﾄ形式	PKOB4
支援するﾃﾞﾊﾞｲｽ形式	全て

11.7. pykitcommander

pykitcommanderはNanoﾃﾞﾊﾞｯｶﾞを使うｷｯﾄで走行する応用とのやり取りを管理します。

何故それを使うのか

MCUに応用を読み込んで仮想ｼﾘｱﾙ ﾎﾟｰﾄを使ってそれと通信することを望む時にpykitcommanderを使ってください。

表11-7. 要約

性質	値
ﾀﾞｳﾝﾛｰﾄﾞ ﾘﾝｸ	pypi.org/project/pykitcommander/
ｿｰｽ ｺｰﾄﾞ	github.com/microchip-pic-avr-tools/pykitcommander
文書資料	microchip-pic-avr-tools.github.io/pykitcommander/
ｺﾏﾝﾄﾞ行ｲﾝﾀｰﾌｪｰｽ	いいえ
ﾗｲﾌﾞﾗﾘ	はい
支援するｷｯﾄ形式	様々
支援するﾃﾞﾊﾞｲｽ形式	全て

12. ピン配置参照基準

表12-1. NanoﾃﾞﾊﾞｯｶﾞSAMD21ﾋﾟﾝ配置

32QFNﾋﾟﾝ	名前	説明
1	CDC_TX	CDC UART送信
2	CDC_RX	CDC UART受信
3	Power supply adjust	電源調整用ｱﾅﾛｸﾞ出力
4	VTG_ADC	目的対象電圧測定用ｱﾅﾛｸﾞ入力
5	DBG0_RX	ﾃﾞﾊﾞｯｸﾞ ｲﾝﾀｰﾌｪｰｽ ﾃﾞｰﾀ用受信信号
6	(予約)	接続しないでください。
7	DBG0_TX	ﾃﾞﾊﾞｯｸﾞ ｲﾝﾀｰﾌｪｰｽ ﾃﾞｰﾀ用送信信号
8	DBG1	ﾃﾞﾊﾞｯｸﾞ ｲﾝﾀｰﾌｪｰｽ用ｸﾛｯｸ
9	VDDANA	SAMD21用ｱﾅﾛｸﾞ電源
10	GNDANA	SAMD21用ｱﾅﾛｸﾞ接地
11	DBG2_CTRL	DBG2(DGI GPIO)用基準移転部方向
12	DBG2	DBG2(DGI GPIO)用ﾃﾞｰﾀ
13	Power supply enable	電源切り換え用許可信号
14	VBUS_ADC	USB VBUS測定用ｱﾅﾛｸﾞ入力
15	DBG0_CTRL	DBG0(ﾃﾞﾊﾞｯｸﾞ ﾃﾞｰﾀ)用基準移転部方向
16	DBG1_CTRL	DBG1(ﾃﾞﾊﾞｯｸﾞ ｸﾛｯｸ)用基準移転部方向
17	DBG3 $\overline{\text{RESET}}$ $\overline{\text{MCLR}}$	DBG3($\overline{\text{RESET}}$,$\overline{\text{MCLR}}$)用開放ﾄﾞﾚｲﾝ制御
18	VTG_EN	目的対象電圧用許可切り換え
19	CDC_RX_CTRL	CRC RX用基準移転部方向
20	VOFF	外部電圧方向
21	CDC_TX_CTRL	CRC TX用基準移転部方向
22	ID_SYS	IDｼｽﾃﾑ
23	USB_DM	USBﾃﾞｰﾀ
24	USB_DP	USBﾃﾞｰﾀ
25	BOOT	ﾌﾞｰﾄﾛｰﾀﾞ移行強制
26	$\overline{\text{RESET}}$	Nanoﾃﾞﾊﾞｯｶﾞ ﾘｾｯﾄ
27	LED	状態LED
28	GND	接地
29	VDDCORE	SAMD21ｺｱ雑音分離
30	VDDIN	SAMD21 VDD
31	SWCLK	Nanoﾃﾞﾊﾞｯｶﾞのﾃﾞﾊﾞｯｸﾞ用SWCLK
32	SWDIO	Nanoﾃﾞﾊﾞｯｶﾞのﾃﾞﾊﾞｯｸﾞ用SWDIO

13. Nanoﾃﾞﾊﾞｯｶﾞ ﾌｧｰﾑｳｪｱ

13.1. ﾌｧｰﾑｳｪｱ一括

Nanoﾃﾞﾊﾞｯｶﾞのﾌｧｰﾑｳｪｱは開放ｿｰｽではありません。これのIntel HEXﾌｧｲﾙはPKOB nano支援下のMicrochip一括貯蔵庫でMicrochipによって公開されたﾂｰﾙ一括内に含まれます。

各ﾂｰﾙ一括内で、nedbg_fw.zipを含むfirmware副ﾌｫﾙﾀﾞを見つけるでしょう。avrtools_fw.xmlﾌｧｲﾙはnedbg.hex HEXﾌｧｲﾙについての情報を提供します。

独立型更新

Nanoﾃﾞﾊﾞｯｶﾞのﾌｧｰﾑｳｪｱを更新するのにpydebuggerupgradeを使ってください。

助言: 最新公開のﾌｧｰﾑｳｪｱを自動的に得るには次の命令を使ってください。

```
pydebuggerupgrade latest
```

助言: 一括版x,y,zのﾌｧｰﾑｳｪｱを自動的にﾀﾞｳﾝﾛｰﾄﾞしてｲﾝｽﾄｰﾙするには次の命令を使ってください。

```
pydebuggerupgrade <x.y.z>
```

IDE更新

最新のNanoﾃﾞﾊﾞｯｶﾞのﾌｧｰﾑｳｪｱ一括をﾀﾞｳﾝﾛｰﾄﾞするのにMPLAB XとVS Code用MPLABﾂｰﾙで一括管理部(Pack Manager)を使ってください。ﾌｧｰﾑｳｪｱはIDEがｷｯﾄに接続する時に自動的に更新されます。

注: ﾌﾟﾛｼﾞｪｸﾄ特性で特定のﾂｰﾙ一括または"latest(最新)"のどちらかを必ず選んでください。

旧型更新

Microchip Studioは同梱されたﾌｧｰﾑｳｪｱを含むだけで、これはもはや更新されません。Nanoﾃﾞﾊﾞｯｶﾞのﾌｧｰﾑｳｪｱを更新するには独立型更新部またはatfw.exeを使ってください。

注: Microchip StudioはNanoﾃﾞﾊﾞｯｶﾞを自動的に最新の同梱ﾌｧｰﾑｳｪｱへ降格するかもしれません。この動きを防ぐにはMicrochip Studioﾌﾟﾛｸﾞﾗﾑ ﾌｧｲﾙ ﾌｫﾙﾀﾞ内のfirmware.zipﾌｧｲﾙを削除または改名してください。

次のようにMicrochip Studioからatfw.exeを使ってください。

助言: ﾂｰﾙ一括から抽出したzip書庫を使ってﾌｧｰﾑｳｪｱを更新するには次の命令を使ってください。

```
atfw.exe -a <zipfile>
```

13.2. 改訂履歴

注: 改訂はこの表で10進数で指定されますが、いくつかのIDEとﾂｰﾙはそれを16進数で報告します。

表13-1. Nanoﾃﾞﾊﾞｯｶﾞ ファームウェアの改訂履歴

改訂	ツール一括	公開日付	参照記号	主な修正と機能
事前1.16	–	2018～2019	FW5G-485	内部的公開とStudio/MPLABでの同梱
1.16	1.0.33	2020年3月	FW5G-516	ツール一括初回公開
1.17		2020年3月	FW5G-620	• SAM-IoT追加
1.20	1.3.164	2020年6月	FW5G-708	• SAM-IoTに関する修正
1.21	1.5.210	2020年9月	FW5G-744	• MSC – CDC橋渡し追加
1.22	1.7.295	2021年3月	FW5G-880	• 複数のCDC修正
1.23	1.9.446	2021年12月	FW5G-1086	• AVR EA支援追加 • AVRﾋｭｰｽﾞ保護に対する修正
1.25	1.10.488	2022年5月	FW5G-1133	• CDC中断(BREAK)用支援追加
1.27	1.11.554	2022年12月	FW5G-1192	• AVR NVM P:4追加 • dsPIC33用支援追加
1.29	1.12.711	2023年10月	FW5G-1250	• AVR NVM P:4に対する修正 • AVR BOOTROW用支援追加 • PDIU保護選別を追加
1.30	1.13.715	2023年10月	FW5G-1306	• 前の1.29 HEXﾌｧｲﾙの問題に対する緊急修正
1.31	1.14.751	2024年4月	FW5G-1323	• UF2解析部追加
1.32	1.16.876	2025年3月	FW5G-1362	• AVR-SD UPDIﾊﾝﾄﾞｼｪｰｸ
1.33	1.17.969	2025年9月	FW5G-1459	• dsPIC33A支援と修正 • DAP_SWI_ﾋﾟﾝ ﾀｲﾐﾝｸﾞ修正
1.34	1.18.x	2025年12月	FW5G-1580	• SWDﾀｲﾐﾝｸﾞ改善

助言: 参照記号はMicrochipｻﾎﾟｰﾄに問い合わせする時に使うことができます。

14. 文書改訂履歴

表14-1. 文書改訂

文書改訂	日付	注釈
A	2025年12月	初版文書公開

Microchip情報

商標

“Microchip”の名称とﾛｺﾞ、“M”のﾛｺﾞ、それと他の名称、ﾛｺﾞ、商標は米国や他の国に於けるMicrochip Technology Incorporatedまたはその系列会社や子会社の登録または未登録の商標です(“Microchip商標”)。Microchip商標に関する情報はhttps://www.microchip.com/en-us/about/legalinformation/microchip-trademarksで見つけることができます。

法的通知

この刊行物と契約での情報は設計、試験、応用とのMicrochip製品の統合を含め、Microchip製品でだけ使えます。他の何れの方法でのこの情報の使用はこれらの条件に違反します。ﾃﾞﾊﾞ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。追加支援については最寄りのMicrochip営業所にお問い合わせ頂くか、www.microchip.com/en-us/support/design-help/client-support-servicesで追加支援を得てください。

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ﾃﾞﾊﾞ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。

Microchipﾃﾞﾊﾞｲｽ ｺｰﾄﾞ保護機能

Microchip製品での以下のｺｰﾄﾞ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ﾃﾞｰﾀｼｰﾄに含まれる仕様に合致します。
- Microchipは動作仕様内で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はその知的所有権を尊重し、積極的に保護します。Microchip製品のｺｰﾄﾞ保護機能を侵害する試みは固く禁じられ、ﾃﾞｼﾞﾀﾙ ﾐﾚﾆｱﾑ著作権法に違反するかもしれません。
- Microchipや他のどの半導体製造業者もそれのｺｰﾄﾞの安全を保証することはできません。ｺｰﾄﾞ保護は製品が”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。ｺｰﾄﾞ保護は常に進化しています。Microchipは当社製品のｺｰﾄﾞ保護機能を継続的に改善することを約束します。

本使用者の手引きはMicrochip Nanoﾃﾞﾊﾞｯｶﾞ手引書(DS50003996A-2025年12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。

製品頁リンク

ATmega1608, ATmega1609, ATmega3208, ATmega3209, ATmega4808, ATmega4809, ATmega808, ATmega809, ATSAM4E16E, ATSAM4LC4C, ATSAM4LC8C, ATSAM4N16C, ATSAM4SD32C, ATSAMC20E15A, ATSAMC20E16A, ATSAMC20E17A, ATSAMC20E18A, ATSAMC20G15A, ATSAMC20G16A, ATSAMC20G17A, ATSAMC20G18A, ATSAMC20J15A, ATSAMC20J16A, ATSAMC20J17A, ATSAMC20J18A, ATSAMC20N17A, ATSAMC20N18A, ATSAMC21E15A, ATSAMC21E16A, ATSAMC21E17A, ATSAMC21E18A, ATSAMC21G15A, ATSAMC21G16A, ATSAMC21G17A, ATSAMC21G18A, ATSAMC21J15A, ATSAMC21J16A, ATSAMC21J17A, ATSAMC21J18A, ATSAMC21N17A, ATSAMC21N18A, ATSAMCA1E17A, ATSAMD20E14, ATSAMD20E15, ATSAMD20E16, ATSAMD20E17, ATSAMD20E18, ATSAMD20G14, ATSAMD20G15, ATSAMD20G16, ATSAMD20G17, ATSAMD20G18, ATSAMD20J14, ATSAMD20J15, ATSAMD20J16, ATSAMD20J17, ATSAMD20J18, ATSAMD21E15, ATSAMD21E15L, ATSAMD21E16, ATSAMD21E16L, ATSAMD21E17, ATSAMD21E17L, ATSAMD21E18, ATSAMD21G15, ATSAMD21G16, ATSAMD21G16L, ATSAMD21G17, ATSAMD21G17L, ATSAMD21G18, ATSAMD21J15, ATSAMD21J16, ATSAMD21J17, ATSAMD21J18, ATSAMD51G18A, ATSAMD51G19A, ATSAMD51J18A, ATSAMD51J19A, ATSAMD51J20A, ATSAMD51N19A, ATSAMD51N20A, ATSAMD51P19A, ATSAMD51P20A, ATSAMDA1E14B, ATSAMDA1E15B, ATSAMDA1E16B, ATSAMDA1G14B, ATSAMDA1G15B, ATSAMDA1G16B, ATSAMDA1J14B, ATSAMDA1J15B, ATSAMDA1J16B, ATSAME51G18A, ATSAME51G19A, ATSAME51J18A, ATSAME51J19A, ATSAME51J20A, ATSAME51N19A, ATSAME51N20A, ATSAME53J18A, ATSAME53J19A, ATSAME53J20A, ATSAME53N19A, ATSAME53N20A, ATSAME54N19A, ATSAME54N20A, ATSAME54P19A, ATSAME54P20A, ATSAME70J19, ATSAME70J20, ATSAME70J21, ATSAME70N19, ATSAME70N20, ATSAME70N21, ATSAME70Q19, ATSAME70Q20, ATSAME70Q21, ATSAMG53, ATSAMHA0E14A-B, ATSAMHA0E15A-B, ATSAMHA0E16A-B, ATSAMHA0G14A-B, ATSAMHA0G15A-B, ATSAMHA0G16A-B, ATSAMHA0G17A-B, ATSAMHA1E14A-B, ATSAMHA1E15A-B, ATSAMHA1E16A-B, ATSAMHA1G14A, ATSAMHA1G14A-B, ATSAMHA1G15A, ATSAMHA1G15A-B, ATSAMHA1G16A, ATSAMHA1G16A-B, ATSAMHA1G17A-B, ATSAML10D14A, ATSAML10D15A, ATSAML10D16A, ATSAML10E14A, ATSAML10E15A, ATSAML10E16A, ATSAML11D14A, ATSAML11D15A, ATSAML11D16A, ATSAML11E14A, ATSAML11E15A, ATSAML11E16A, ATSAML21E15B, ATSAML21E16B, ATSAML21E17B, ATSAML21E18B, ATSAML21G16B, ATSAML21G17B, ATSAML21G18B, ATSAML21J16B, ATSAML21J17B, ATSAML21J18B, ATSAML22G16A, ATSAML22G17A, ATSAML22G18A, ATSAML22J16A, ATSAML22J17A, ATSAML22J18A, ATSAML22N16A, ATSAML22N17A, ATSAML22N18A, ATSAMS70J19, ATSAMS70J20, ATSAMS70J21, ATSAMS70N19, ATSAMS70N20, ATSAMS70N21, ATSAMS70Q19, ATSAMS70Q20, ATSAMS70Q21, ATSAMV70J19, ATSAMV70J20, ATSAMV70N19, ATSAMV70N20, ATSAMV70Q19, ATSAMV70Q20, ATSAMV71J19, ATSAMV71J20, ATSAMV71J21, ATSAMV71N19, ATSAMV71N20, ATSAMV71N21, ATSAMV71Q19, ATSAMV71Q20, ATSAMV71Q21, ATtiny1604, ATtiny1606, ATtiny1607, ATtiny1614, ATtiny1616, ATtiny1617, ATtiny1624, ATtiny1626, ATtiny1627, ATtiny202, ATtiny204, ATtiny212, ATtiny214, ATtiny3216, ATtiny3217, ATtiny3224, ATtiny3226, ATtiny3227, ATtiny402, ATtiny404, ATtiny406, ATtiny412, ATtiny414, ATtiny416, ATtiny417, ATtiny424, ATtiny426, ATtiny427, ATtiny804, ATtiny806, ATtiny807, ATtiny814, ATtiny816, ATtiny817, ATtiny824, ATtiny826, ATtiny827, ATtiny828, ATtiny861, AV32DU20, AV32DU28, AVR128DA28, AVR128DA28S, AVR128DA32, AVR128DA32S, AVR128DA48, AVR128DA48S, AVR128DA64, AVR128DA64S, AVR128DB28, AVR128DB32, AVR128DB48, AVR128DB64, AVR16DD14, AVR16DD20, AVR16DD28, AVR16DD32, AVR16DU14, AVR16DU20, AVR16DU28, AVR16DU32, AVR16EA28, AVR16EA32, AVR16EA48, AVR16EB14, AVR16EB20, AVR16EB28, AVR16EB32, AVR16LA14, AVR16LA20, AVR16LA28, AVR16LA32, AVR32DA28, AVR32DA28S, AVR32DA32, AVR32DA32S, AVR32DA48, AVR32DA48S, AVR32DB28, AVR32DB32, AVR32DB48, AVR32DD14, AVR32DD20, AVR32DD28, AVR32DD32, AVR32DU14, AVR32DU20, AVR32DU28, AVR32DU32, AVR32EA28, AVR32EA32, AVR32EA48, AVR32EB14, AVR32EB20, AVR32EB28, AVR32EB32, AVR32EC28, AVR32EC32, AVR32EC48, AVR32LA14, AVR32LA20, AVR32LA28, AVR32LA32, AVR32SD20, AVR32SD28, AVR32SD32, AVR64DA28, AVR64DA28S, AVR64DA32, AVR64DA32S, AVR64DA48, AVR64DA48S, AVR64DA64, AVR64DA64S, AVR64DB28, AVR64DB32, AVR64DB48, AVR64DB64, AVR64DD14, AVR64DD20, AVR64DD28, AVR64DD32, AVR64DU28, AVR64DU32, AVR64EA28, AVR64EA32, AVR64EA48, AVR64EC28, AVR64EC32, AVR64EC48, AVR64SD28, AVR64SD32, AVR64SD48, PIC16F13145, PIC16F13276, PIC16F15244, PIC16F15276, PIC16F17146, PIC16F18076, PIC16F18446, PIC18F16Q20, PIC18F16Q40, PIC18F16Q41, PIC18F47K42, PIC18F47Q10, PIC18F56Q24, PIC18F56Q35, PIC18F56Q71, PIC18F57Q43, PIC18F57Q84, PIC32CK0512GC00064, PIC32CK0512GC00100, PIC32CK0512GC01064, PIC32CK0512GC01100, PIC32CK0512SG00064, PIC32CK0512SG00100, PIC32CK0512SG01064, PIC32CK0512SG01100, PIC32CK1025GC00064, PIC32CK1025GC00100, PIC32CK1025GC01064, PIC32CK1025GC01100, PIC32CK1025GC01144, PIC32CK1025SG00064, PIC32CK1025SG00100, PIC32CK1025SG01064, PIC32CK1025SG01100, PIC32CK1025SG01144, PIC32CK2051GC00064, PIC32CK2051GC00100, PIC32CK2051GC00144, PIC32CK2051GC01064, PIC32CK2051GC01100, PIC32CK2051GC01144, PIC32CK2051SG00064, PIC32CK2051SG00100, PIC32CK2051SG00144, PIC32CK2051SG01064, PIC32CK2051SG01100, PIC32CK2051SG01144, PIC32CM1216JH01032, PIC32CM1216MC00032, PIC32CM1216MC00048, PIC32CM1216PL10028, PIC32CM1216PL10032, PIC32CM1216PL10048, PIC32CM1216PL10064, PIC32CM1602GV00032, PIC32CM1602GV00048, PIC32CM1602GV00064, PIC32CM1602PL10020, PIC32CM1602PL10028, PIC32CM1602PL10032, PIC32CM2532JH00032, PIC32CM2532JH00048, PIC32CM2532JH00064, PIC32CM2532JH00100, PIC32CM2532JH01032, PIC32CM2532JH01048, PIC32CM2532JH01064, PIC32CM2532JH01100, PIC32CM2532LE00048, PIC32CM2532LE00064, PIC32CM2532LE00100, PIC32CM2532LS00048, PIC32CM2532LS00064, PIC32CM2532LS00100, PIC32CM2532LS60048, PIC32CM2532LS60064, PIC32CM2532LS60100, PIC32CM3204GV00032, PIC32CM3204GV00048, PIC32CM3204GV00064, PIC32CM3204JH00032, PIC32CM3204JH00048, PIC32CM3204JH00064, PIC32CM3204PL10020, PIC32CM3204PL10028, PIC32CM3204PL10032, PIC32CM5112GC00048, PIC32CM5112GC00064, PIC32CM5112GC00100, PIC32CM5112SG00048,

PIC32CM5112SG00064, PIC32CM5112SG00100, PIC32CM5164JH00032, PIC32CM5164JH00048, PIC32CM5164JH00064, PIC32CM5164JH00100, PIC32CM5164JH01032, PIC32CM5164JH01048, PIC32CM5164JH01064, PIC32CM5164JH01100, PIC32CM5164LE00048, PIC32CM5164LE00064, PIC32CM5164LE00100, PIC32CM5164LS00048, PIC32CM5164LS00064, PIC32CM5164LS00100, PIC32CM5164LS60048, PIC32CM5164LS60064, PIC32CM5164LS60100, PIC32CM6408JH00032, PIC32CM6408JH00048, PIC32CM6408JH00064, PIC32CM6408MC00032, PIC32CM6408MC00048, PIC32CM6408PL10028, PIC32CM6408PL10032, PIC32CM6408PL10048, PIC32CM6408PL10064, PIC32CX1012BZ24032, PIC32CX1012BZ25048, PIC32CX1025MTC, PIC32CX1025MTG, PIC32CX1025MTSH, PIC32CX1025SG41100, PIC32CX1025SG41128, PIC32CX1025SG60100, PIC32CX1025SG60128, PIC32CX1025SG61100, PIC32CX1025SG61128, PIC32CX2051BZ62132, PIC32CX2051MTC, PIC32CX2051MTC128, PIC32CX2051MTG, PIC32CX2051MTSH, PIC32CX5109BZ31032, PIC32CX5109BZ31048, PIC32CX5109BZ36032, PIC32CX5109BZ36048, PIC32CX5112MTC, PIC32CX5112MTG, PIC32CX5112MTSH, PIC32CX-BZ2, PIC32CX-BZ3, PIC32CX-BZ6, PIC32CZ2051CA70064, PIC32CZ2051CA70100, PIC32CZ2051CA70144, PIC32CZ2051CA80100, PIC32CZ2051CA80144, PIC32CZ2051CA80176, PIC32CZ2051CA80208, PIC32CZ2051CA90100, PIC32CZ2051CA90144, PIC32CZ2051CA90176, PIC32CZ2051CA90208, PIC32CZ2051MC70064, PIC32CZ2051MC70100, PIC32CZ2051MC70144, PIC32CZ4010CA80100, PIC32CZ4010CA80144, PIC32CZ4010CA80176, PIC32CZ4010CA80208, PIC32CZ4010CA90100, PIC32CZ4010CA90144, PIC32CZ4010CA90176, PIC32CZ8110CA80100, PIC32CZ8110CA80144, PIC32CZ8110CA80176, PIC32CZ8110CA80208, PIC32CZ8110CA90100, PIC32CZ8110CA90144, PIC32CZ8110CA90176, PIC32CZ8110CA90208