

ライブラリの範囲

AVR® MCU電動機制御ライブラリは感知器ありまたは感知器なしのどちらかを使ってブラシレス直流(BLDC:BrushLess Direct Current)電動機と永久磁石同期電動機(MPSM:Permanent Magnet Synchronous Motors)を回すための支援を持つAVR EB系デバイス用のライブラリです。このライブラリ インターフェースは電動機指定と電力基板の望む変更、マイクロ コントローラ(MCU)設定、ピン配置機能を提供します。駆動算法はMCU周辺機能の利点を取り、CPUが大きな付随負荷を持たず、メモリの使用と資源の消費を最適化することを保証します。簡単な実行時構成設定と制御に関して使用者層APIが生成されます。

生成されたコードは使用者ハードウェアまたは電力基板、マイクロ コントローラ用アダプタ、Curiosity Nanoを含むMicroshipで入手可能な実演キットで動きます。

- ・ 多相電力基板(MPPB) ([EV35Z86A](#))
- ・ AVR16EB32-CNANO MPPBアダプタ ([EV88N31A](#))
- ・ AVR16EB32 Curiosity Nano ([EV73J36A](#))

この使用者の手引きはインターフェースと生成したコードが使用者に提供する全ての機能を詳しく見ます。

目次

ライブラリの範囲	1
1. 動作環境	3
2. 支援されるMCU	4
3. 現版の機能 (1.2.0版)	5
4. 電動機制御ライブラリ仕様	6
5. インストール	9
6. 構成設定	11
6.1. 構成部操作盤	11
6.2. 応用構築部操作盤	20
6.3. ピン格子操作盤	20
6.4. 資源管理操作盤	22
7. 生成したAPI	23
8. 通信	24
8.1. 端末経由の通信 (printf)	24
8.2. DVRT経由の通信	26
9. 実演	31
10. 手順	32
10.1. 手順1. ハードウェア設定	32
10.2. 手順2. MPLAB®設定	33
10.3. 手順3. AVR® MCC電動機制御ライブラリ設定	37
10.4. 手順4. 書き込み設定	39
10.5. 手順5. 実演指示	39
10.6. 手順6. 実演調整	40
11. 改訂履歴	41
Microchip情報	42
商標	42
法的通知	42
Microchipデバイスコード保護機能	42

1. 動作環境

- MPLAB® X IDE 6.20またはそれ以降版
- MPLAB® XC8コンパイラ 2.50またはそれ以降版
- MPLAB®コर्ट構成部(MCC) 5.5.1またはそれ以降版
- MCC核 5.7.1またはそれ以降版
- Melody核 2.7.1またはそれ以降版
- MPLAB®データ可視器(Data Visualizer)プラグイン 1.3.1665またはそれ以降版
- DFP – Microchip AVR-Ex系デバイス支援 2.10.205またはそれ以降版

2. 支援されるMCU

このライブラリは以下の8ビットAVRデバイスを支援します。

- AVR16EB20
- AVR16EB28
- AVR16EB32
- AVR32EB20
- AVR32EB28
- AVR32EB32

3. 現版の機能 (1.2.0版)

- ・ インターフェースと構成設定ファイルでの電動機仕様 (BLDC)
- ・ 台形駆動
- ・ 正弦状駆動 (空間ベクトル パルス幅変調(SVPWM:Space Vector Pulse-Width Modulation)特性、サドル特性)
- ・ ホール感知器帰還支援
- ・ 逆起電力(BEMF:Back-Electromotive Force)感知 (台形動作専用)
- ・ 電動機駆動(回転子-固定子)同期 (開路)
- ・ 位相進み選択
- ・ 開始と停止の傾斜
- ・ 実時間で要素を作画するためのMPLABデータ可視器走行時(DVRT:Data Visualizer Run Time)通信支援
- ・ 速度調整の閉路制御のための固定要素での比例積分(PI:Proportional-Integral)算法
- ・ ハードウェア頂点過電流保護(頂点OCP)とソフトウェア平均過電流保護(平均OCP)、過電圧保護(OVP)、低電圧保護(UVP)、過熱保護(OTEMP)、失速検出とホール感知器切断に対する障害支援
- ・ 電流、電圧バス、温度、可変抵抗器の走行時アナログ測定
- ・ 15kHz～45kHzの可変切り替え周波数
- ・ 低速感知器制御 (電動機の公称速度の5%)
- ・ 始動でのホール監視器自動検出算法

4. 電動機制御ライブラリ仕様

- ・ 支援電動機形式: 3相BLDCとPMSM
- ・ 帰還形式: BEMF測定での感知器なし、またはホール感知器
- ・ PWM駆動周波数: 15kHz～45kHzで構成設定可能
- ・ 駆動変調能力: 塊整流(台形)、正弦状(正弦、SVPWM、サトル動作)
- ・ 駆動と帰還の同期方法: 位相固定化閉路(PLL)近似
- ・ 電氣的RPM変調周波数: 20～2000Hz
- ・ 同期消失最低速度: BEMFの場合は電動機RPMの15%、ホール感知器の場合は20Hz
- ・ 傾斜API: 2点だけを持つ単一の定率増加と定率減少(多点傾斜なし)
- ・ 最小/最大間加速時間: 1.6384秒 (16ビット速度から1ビット増加)
- ・ 速電流測定能力: 比較器頂点起動とA/D変換器(ADC)累積平均
- ・ ADC分解能: 12ビット単一測定/16ビット累積平均
- ・ 沈黙時間制御: 50ns段階の8ビット制御で0～12750ns
- ・ 段階変更での開路応答時間: 300～500ms
- ・ 段階変更での閉路応答時間: 500～750ms
- ・ 障害能力: 過電圧保護(OVP)、低電圧保護(UVP)、温度監視、過電流保護(OCF) – 頂点と平均
- ・ 障害応答時間(比較器を使う頂点電流): 最小100～150ns (図4-1をご覧ください。)
- ・ 障害応答時間(ADC使用): アナログ濾波器(滑らかな値を得るためのソフトウェア濾波)許可で30～35ms、アナログ濾波器禁止で1.5～2ms (図4-2をご覧ください。)
- ・ 帰還応答時間: 1駆動PWM周期 (20kHz駆動に対して50μs)
- ・ 正弦状動作での総割り込み窓占有時間: 約23μs (図4-3をご覧ください。)
- ・ 台形動作での総割り込み窓占有時間: 区間間で約19μsまたは区間切り替えで30μs (図4-4をご覧ください。)
- ・ 停止/再始動間の安全遅延時間: 2で割った減少期間[ms]が減少終了後の停止/再始動間の安全時間
- ・ 位相進み制御: 固定で0～90°
- ・ ホール感知器検出遅延: 検出に対して1駆動PWM周期、調整に対して30～120の電氣的角度
- ・ BEMF検出遅延: 検出に対して1駆動PWM周期、調整に対して30～120の電氣的角度
- ・ 正弦歪能力: なし
- ・ 回生制動能力: なし
- ・ 電動機が回っている(風車状態)間の開始能力: なし
- ・ 弱め界磁(磁束)能力: なし

図4-1. 頂点電流起動障害保護応答時間

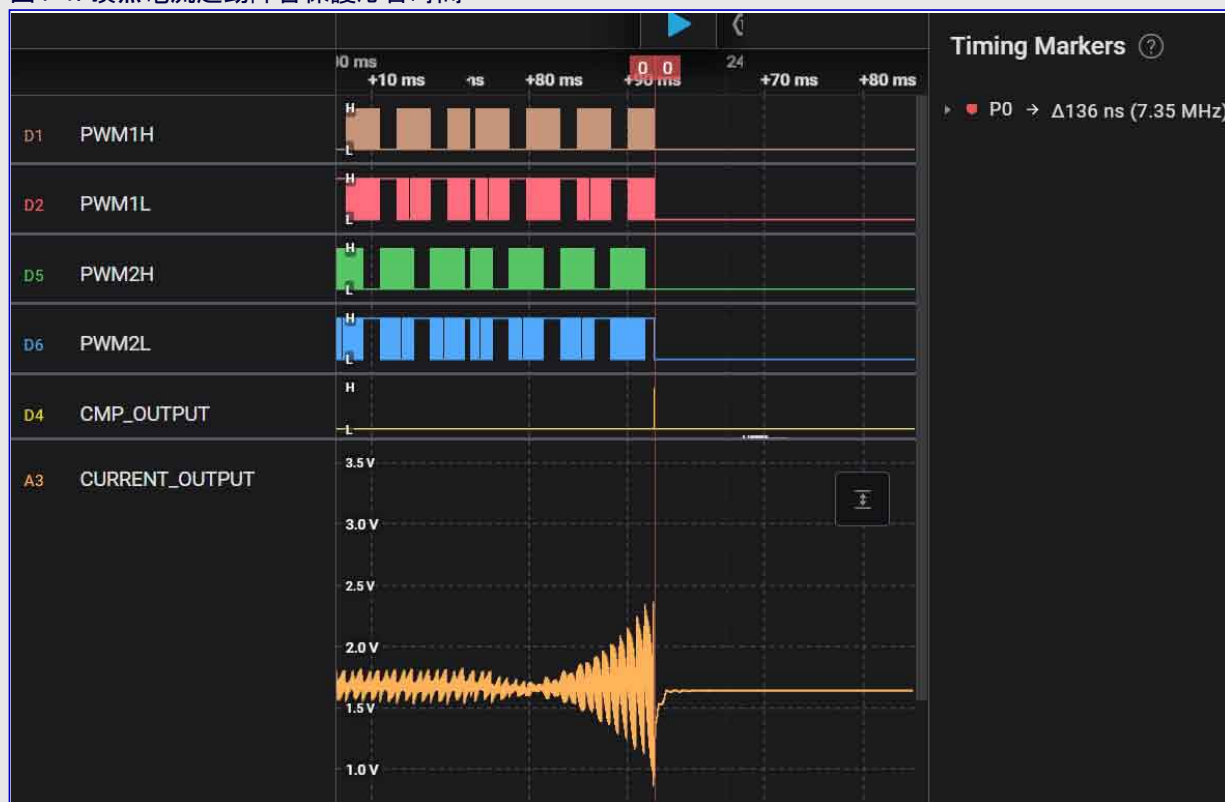


図4-2. 過電圧障害保護応答時間

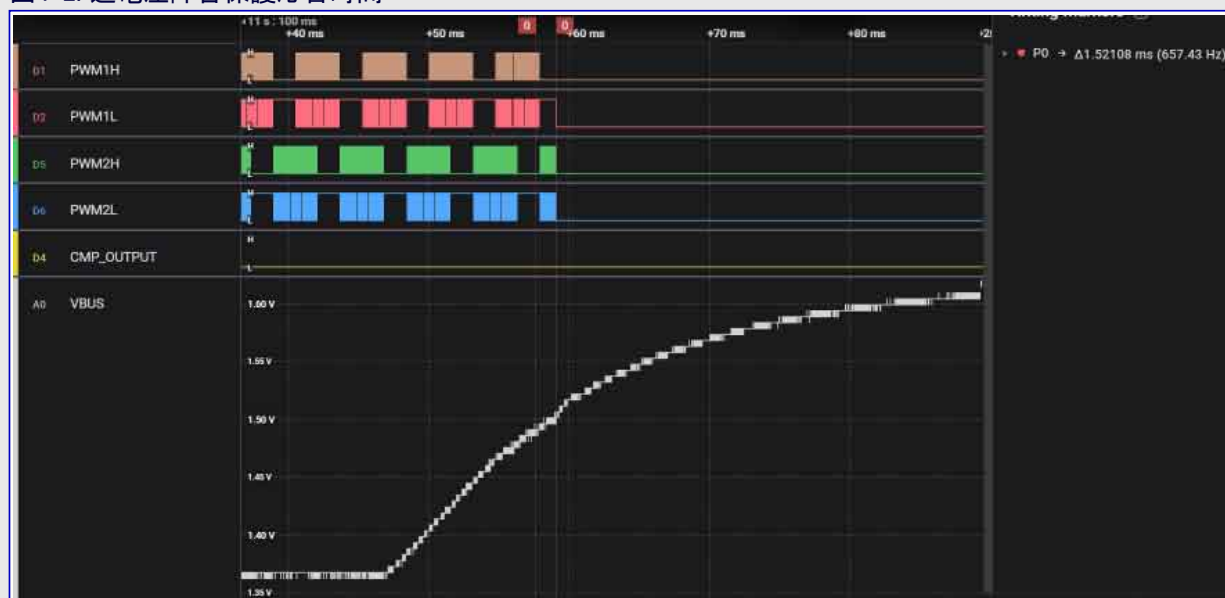


図4-3. 正弦状駆動動作での割り込み窓占有

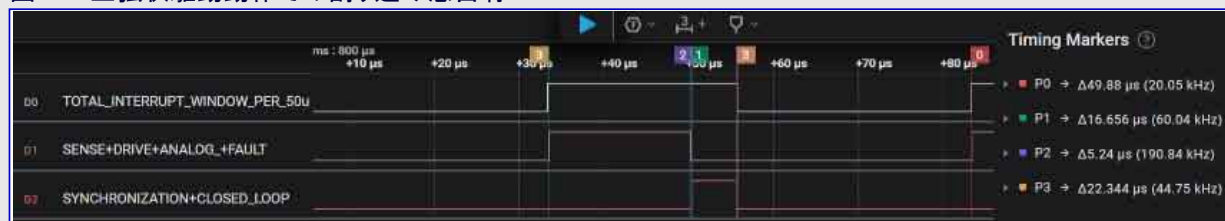
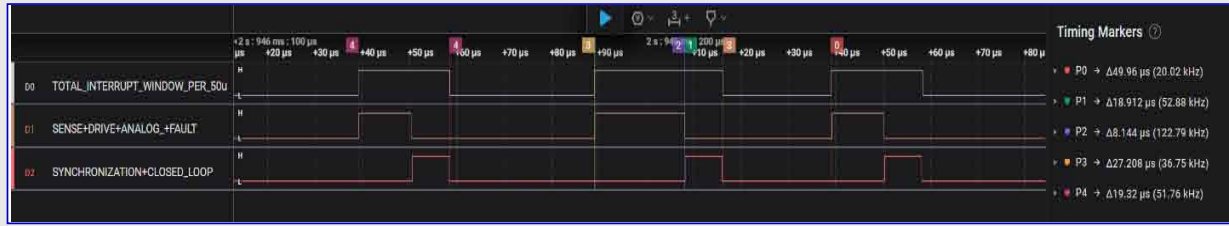
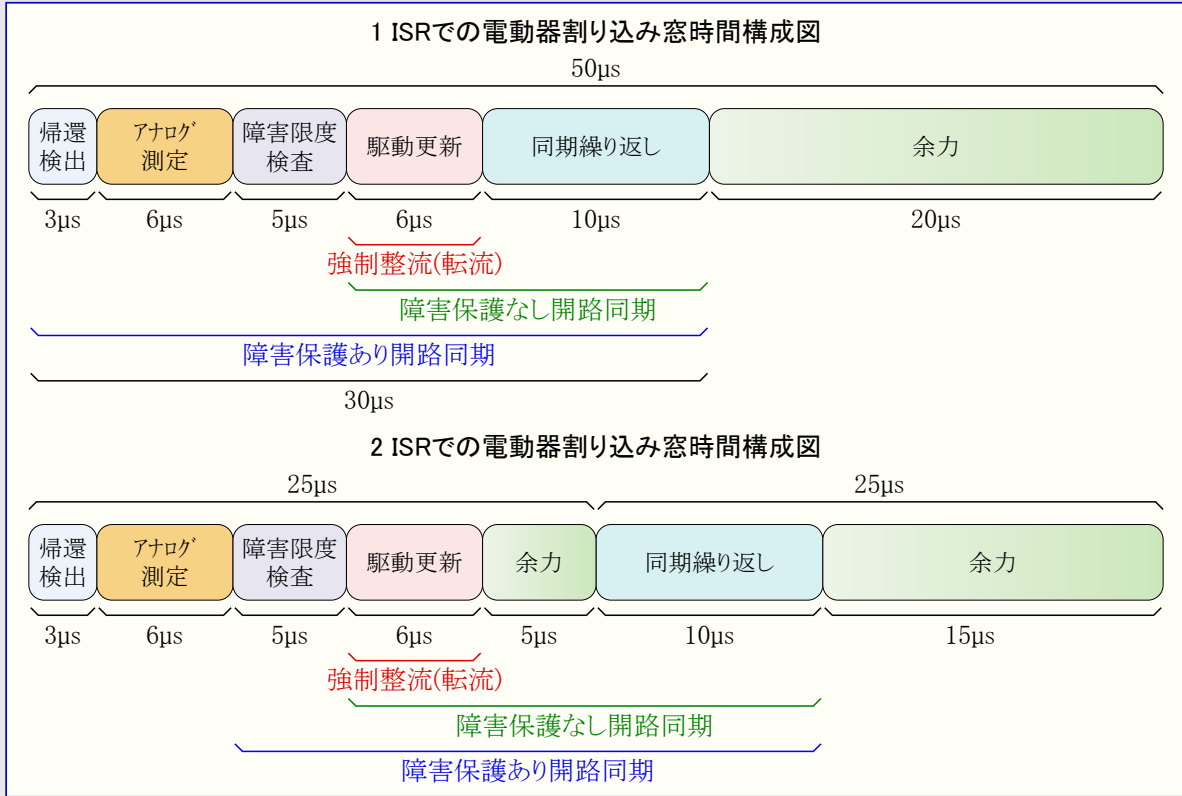


図4-4. 台形駆動動作での割り込み窓占有



MOSFET切り替え周波数に基づき、割り込み窓は2つの連続する割り込み処理ルーチン(ISR)に分割されるか、または単一のISRで実行されます。PWM周波数が20kHzを超えると、計算は2つのISRの間に行われ、20kHz以下の時は1つのISRだけです。20kHzと40kHzに対する例を下図をご覧ください。

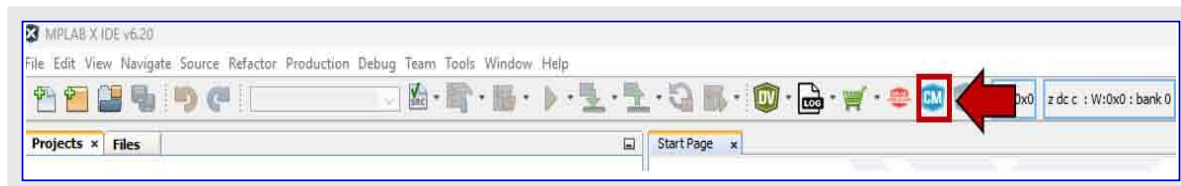
図4-5. 電動機割り込み窓時間構成図



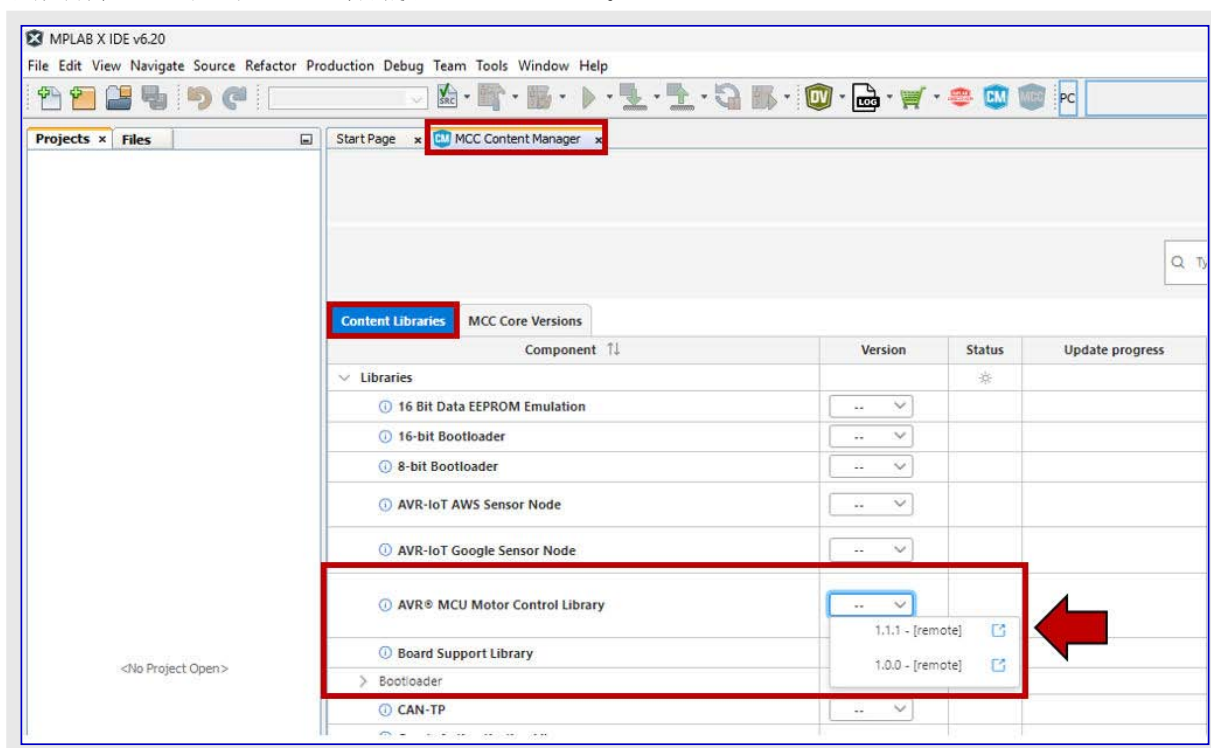
5. インストール

本章はMPLABコード構成部(MCC:MPLAB Code Configurator)版用インストール手順、組み込み版の使い方、最初の書き込みとデバッグの方法を提示します。

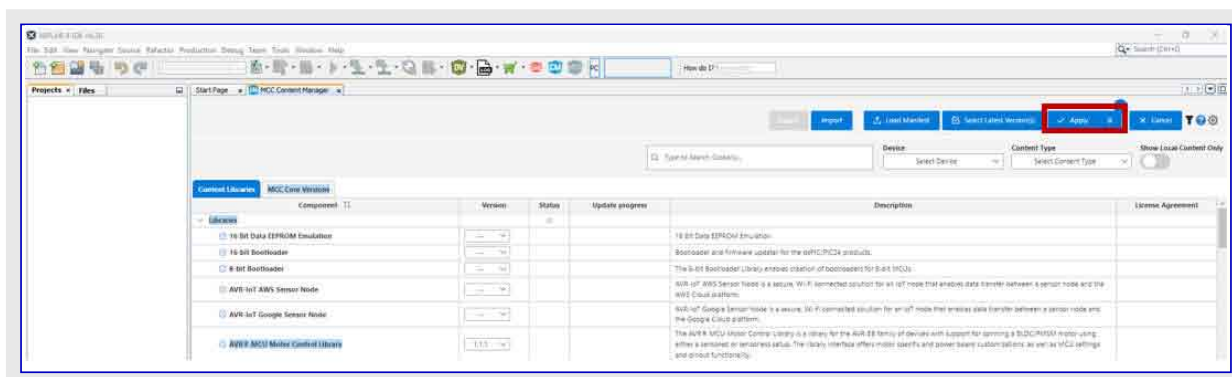
1. MPLAB X IDE 6.20またはそれ以降版を開いてください。
2. **Content Manager**(内容管理部)を開いてください。



3. **MCC Content Manager**(MCC内容管理部)タブで**Content Libraries**(内容ライブラリ)下で”**AVR MCU Motor Control Library**(AVR MCU電動機制御ライブラリ)”を見つけて最新版を選んでください。

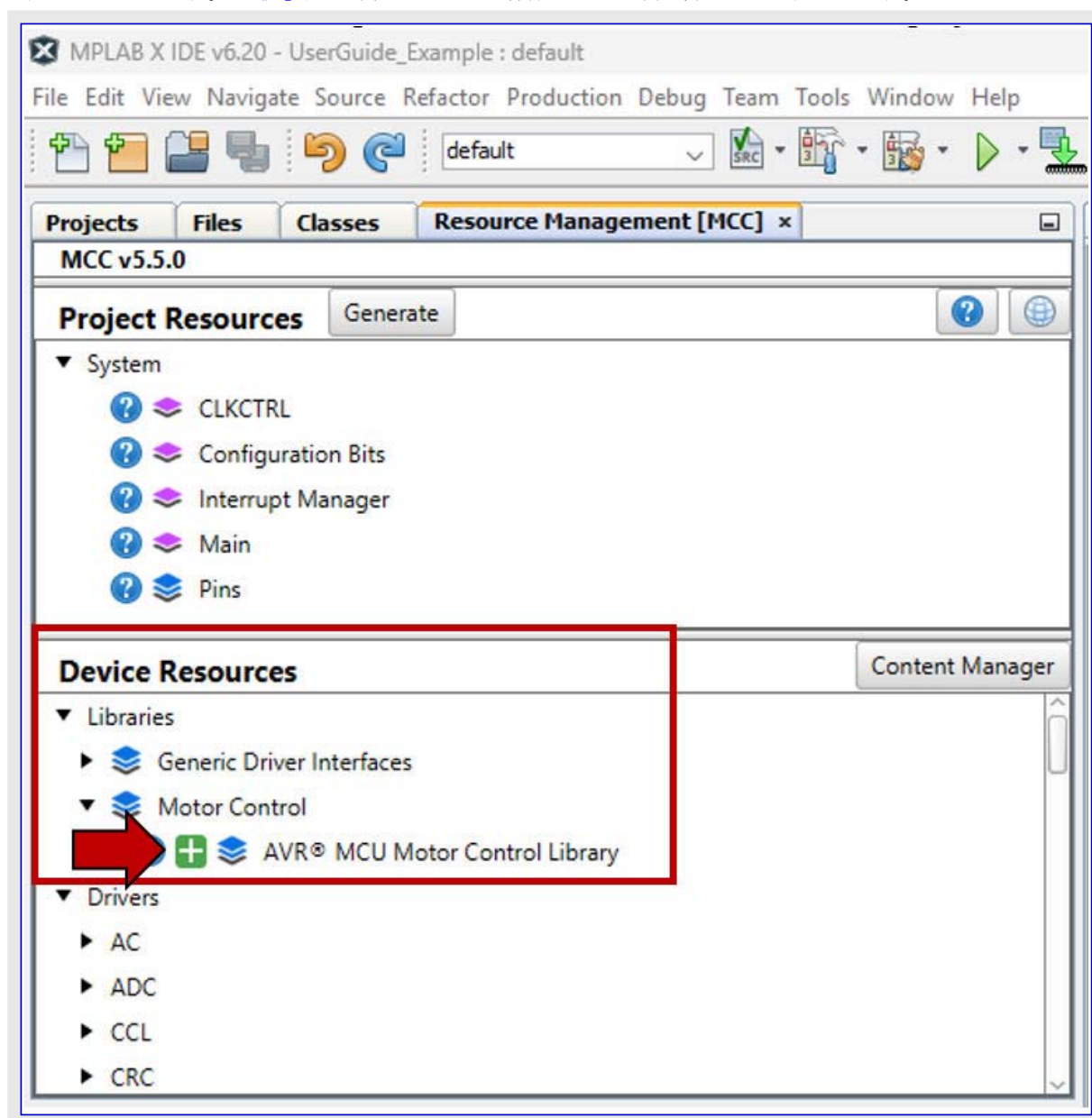


4. ライブラリとMCCとの依存関係をインストールするために**Apply**(適用)鈕をクリックしてください。



5. インストールは今や完了です。

支援されるMCUを使って新規プロジェクトを作成してMCCを開いた後、このライブラリが利用可能で、Device Resources(デバイス資源)ウィンドウから追加することができます。「実演」章は新規プロジェクトの作成とこの手順を得る方法を記述します。



6. 構成設定

下図はライブラリをプロジェクトに追加した後で使用者が会う最初のインターフェースを提示します。このインターフェースは以下の構成部分から成ります。

- **Resource Management Panel(資源管理操作盤):**
資源管理領域は樹状と平面の2つの視野を持ちます。両方共ソフトウェア/周辺機能構成部品と現在のプロジェクト構成設定に対して選んだ構成部品の完全な一覧へのアクセスを提供します。
- **Application Builder Panel(応用構築部操作盤):**
使われる資源と共に構造化した関連性管理は構成部品の関連する依存性とプロジェクトの内容の明確な可視化を提供します。
- **Composer Panel(構成部操作盤):**
周辺機能、ライブラリ、他の外部構成部品がProject Resource(プロジェクト資源)領域から選ばれると、Composer(構成部)領域で対応する構成設定GUIが表示されます。
- **Pins Package Panel(ピン外圍器操作盤):**
画像視野でピンを表示して外圍器図上にカーソルを浮かせている時にマウスの輪をスクロールすることによって拡大/縮小機能を提供します。
- **Pin Grid Panel(ピン格子操作盤):**
施錠と解錠の機能を持つ構成表形式でピンを表示



6.1. 構成部操作盤

構成部操作盤がライブラリ機能に関連する全ての構成設定を含む一方で、ピン格子操作盤は正確なピン機能と位置を含みます。個別に取り上げると、電動機制御ライブラリの各部門は以下の任意選択を持ちます。

1. General Settings(全般設定)

全般設定は駆動と帰還の動作形態用の制御設定を保持すると同時に実演例の生成に使用者に許します。

- **Commutation Feedback**(整流(転流)帰還): 感知器ありまたは感知器なしの帰還算法間の電動機同期供給元に対する選択

- **Feedback Source**(帰還元): 帰還それ自身の供給元の変更を許します。
感知器なし帰還については、この版での供給元としてBEMFだけが受け入れられるため、この任意選択は灰色表示されます。(感知器ありの場合、)現版に対してはホール(感知器)に固定化されますが、将来版で直交符号器と誘導型感知器が任意選択に含まれるでしょう。
- **Drive Mode**(駆動様式): 実際のPWM低周波数駆動変調様式は6段階転流としても知られる台形駆動と正弦状駆動から選ぶことができます。
正弦状駆動変調は標準正弦、空間ベクトル変調(SVM)、サトル動作の3つの異なる参照表(LUT)生成任意選択を持ちます。

- **Generate Example**(例生成): この任意選択は抵抗器速度制御で電動機を回すのに必要なAPIを使う例応用の生成を許します。初期化と繰り返し論理の関数がmain.cファイル実演で呼ばれます。

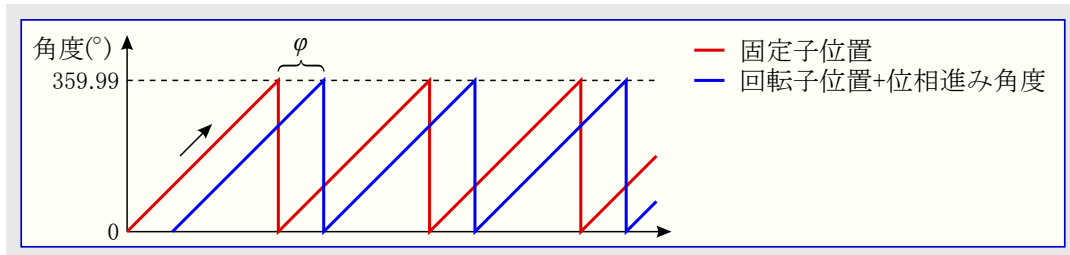
2. Motor Specifications(電動機仕様)

この部分では制御と速度測定に使われる電動機に関連する特性を望む値に設定することができます。

- **Pole pairs(極対)**: この値は電動機のデータシートの「極数」部分から取られ、極対値を得るにはその値を2で割ってください。極対情報は速度測定に使われるため、例えばこれが不正に書かれたとしても電動機は回りますが、測定した速度は正確ではありません。

▼ Motor Specifications			
Pole Pairs ⓘ	1 ≤ 4	↕	≤ 28
Phase Advance [°] ⓘ	0 ≤ 15	↕	≤ 90
Start-up Current [A] ⓘ	0 ≤ 2.5	↕	≤ 10
Kv [V/RPM] ⓘ	0 ≤ 0.007	↕	≤ 5
Phase-to-Phase Resistance [Ω] ⓘ	0 ≤ 0.4	↕	≤ 100

- **Phase Advance [°](位相進み)**: 0~90の電氣的度の範囲にすることができます。感知算法は回転子位置を推定して固定子(駆動界磁)と回転子(電動機界磁)間の差を計算します。(電氣的角度での)結果は駆動界磁が電動機界磁とどれほど位相がずれているを表します。この差に基づき、駆動速度は固定子と回転子間で一定の相移動を保つように更新されます。相移動が小さすぎる場合、トルクは同期を保つのに充分でないかもしれません。相移動が大きすぎる場合、同期も失われます。最適値は充分なトルクが生成されて電力消費が効率的なことを保証するように決定されます。



- **Start-up Current [A](始動電流)**: 始動段階の間に電動機相に供給される電流がAで与えられます。供給される電圧に基づき、これはソフトウェアでデューティサイクルに変換されます。供給電圧が始動電圧未満の場合、応用は誤りメッセージを示して電動機速度上昇は失敗します。

▼ Motor Specifications			
Pole Pairs ⓘ	1 ≤ 4	↕	≤ 28
Phase Advance [°] ⓘ	0 ≤ 15	↕	≤ 90
Start-up Current [A] ⓘ	0 ≤ 2.5	↕	≤ 10
Kv [V/RPM] ⓘ	0 ≤ 0.007	↕	≤ 5
Phase-to-Phase Resistance [Ω] ⓘ	0 ≤ 0.4	↕	≤ 100

- **Kv [V/RPM](Kv)**: このライブラリでこれは(BEMF定数としても知られる)電動機速度定数の逆数を表し、電圧(V)/分当たりの回転数(RPM)で測定されます。電動機のデータシートでこの値を見つけることができます。これは電動機が負荷なしで発電機として駆動される時に測定することができます。端末で測定した電圧は電動機/発電機のKv当たりのRPMに完全に比例します。

例: 電動機を100RPMで回す間での1Vの測定はKv=100になる一方で、ライブラリでそれは1/100で、故に使用者は0.01を入力しなければなりません。

▼ Motor Specifications		
Pole Pairs ⓘ	1 ≤ 4	≤ 28
Phase Advance [°] ⓘ	0 ≤ 15	≤ 90
Start-up Current [A] ⓘ	0 ≤ 2.5	≤ 10
Kv [V/RPM] ⓘ	0 ≤ 0.007	≤ 5
Phase-to-Phase Resistance [Ω] ⓘ	0 ≤ 0.4	≤ 100

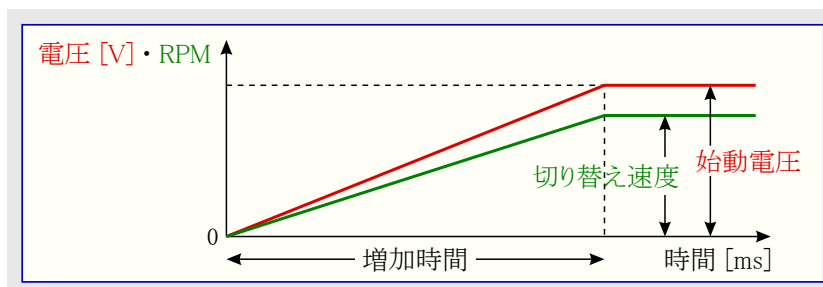
- Phase-to-Phase Resistance [Ω](相間抵抗): 電動機のどれかの2相間で測定された(Ωでの)電氣的抵抗。

▼ Motor Specifications		
Pole Pairs ⓘ	1 ≤ 4	≤ 28
Phase Advance [°] ⓘ	0 ≤ 15	≤ 90
Start-up Current [A] ⓘ	0 ≤ 2.5	≤ 10
Kv [V/RPM] ⓘ	0 ≤ 0.007	≤ 5
Phase-to-Phase Resistance [Ω] ⓘ	0 ≤ 0.4	≤ 100

General Settings(全般設定)⇒Commutation Feedback(整流(転流)帰還)で感知器あり動作が選ばれると、電動機仕様設定でより多くの選択が利用可能です。

3. Start-up Specifications(始動仕様)

この部分では0から電動機始動電圧要素で決定された値まで増加するのと減少状態で0に戻るのに(PWMデューティサイクルによって特定される)駆動電圧に対して取られる時間を望むように変更することができます。この要素は電動機始動ルーチンと結び付けられ、強制整流(転流)から駆動と電動機同期までの速度切り替え遷移を含みます。



- Ramp-up Time [ms](増加時間): 0Vから望む値まで電圧を増加するのと0 RPMから望む切り替え速度まで駆動速度を増加するための始動算法に提供されるms(m秒)での時間。

▼ Start-up Specifications		
Ramp-up Time [ms] ⓘ	10 ≤ 1000	≤ 10000
Ramp-down Time [ms] ⓘ	10 ≤ 1000	≤ 10000
Start-up Speed [RPM] ⓘ	10 ≤ 400	≤ 5000
Forced Commutation ⓘ	☐	

- **Ramp-down Time [ms]**(減少時間): 現在値から0Vまで電圧を減少するのと、また現在のRPMから0 RPMまで駆動速度を減少するための電動機停止算法に提供されるms(m秒)での時間。

Start-up Specifications

Ramp-up Time [ms] ⓘ 10 ≤ 1000 ≤ 10000

Ramp-down Time [ms] ⓘ 10 ≤ 1000 ≤ 10000

Start-up Speed [RPM] ⓘ 10 ≤ 400 ≤ 5000

Forced Commutation ⓘ ☐

- **Start-up Speed [RPM]**(始動速度): 0から目的速度まで回転を上げる増加算法によって目的RPM速度として使われます。これは強制整流(転流)と同期繰り返し開始間を切り換える速度です。感知器なし動作が使われる場合、この要素は電動機の公称速度の最低15～20%に設定されなければなりません。

Start-up Specifications

Ramp-up Time [ms] ⓘ 10 ≤ 1000 ≤ 10000

Ramp-down Time [ms] ⓘ 10 ≤ 1000 ≤ 10000

Start-up Speed [RPM] ⓘ 10 ≤ 400 ≤ 5000

Forced Commutation ⓘ ☐

- **Forced Commutation**(強制整流(転流)): この機能が許可されると、駆動は電動機の帰還と同期しません。代わりに、望む電圧と速度に留まることを強制されます。

Start-up Specifications

Ramp-up Time [ms] ⓘ 10 ≤ 1000 ≤ 10000

Ramp-down Time [ms] ⓘ 10 ≤ 1000 ≤ 10000

Start-up Speed [RPM] ⓘ 10 ≤ 400 ≤ 5000

Forced Commutation ⓘ ☐

4. Pulse-Width Modulation Settings(パルス幅変調設定)

この部分では駆動するPWM信号、開始する主周波数、出力対への沈黙時間挿入構成設定を望むように設定できます。これは電動機制御を抽象化するための低位周辺構成設定を許します。

- **Frequency**(周波数): どれかの駆動様式に対するPWM周波数で、15.0～45.0kHzで使用者選択可能です。

- **Low-Side Dead-time [ns]**(Low側沈黙時間)と**High-Side Dead-time [ns]**(High側沈黙時間): High側とLow側のMOSFETを駆動する補完信号は電流貫通とハードウェア損傷を防ぐために両信号がLowの時の沈黙時間が必要です。この値はns(n秒)で、段階はMCU主クロック周期に依存します(例えば、20MHz主クロックに対してそれらは50nsの沈黙時間段階です)。

- **Output Pair A/B/C**(出力対A/B/C): ライブラリは駆動する補完対信号が違うポートピンに配置されるのを許します。この任意選択はMCUの入出力(I/O)で与えられた対の変更を許します。駆動信号の位置を変更する能力は制御基板のハードウェア設計中の柔軟性を提供します。

5. Sensing Settings(感知設定)

この部分では感知層に対する構成部品の物理的な値を望む値にすることができます。

- **Shunt Resistor [Ω]**(電流検出抵抗): Ω で与えられる電流検出抵抗は電力基板に依存します。
- **Current Amplifier Gain**(電流増幅器利得): 電流検出増幅器利得は電力基板に依存します。
- **Voltage Reference [V]**(電圧参照基準): ADC電圧参照基準はマイクロコントローラのVDD値でなければなりません。これはADC測定計算と解釈に使われます。この値の変更はそれがVDDに設定されるため、ADCの参照基準を変更しません。
- **Voltage Divider**(分圧器): VBUS測定用直列接続抵抗の分圧係数。
- **Temperature Offset [mV]**(温度変位): 温度感知器に対して加えられるADC測定点変位は電力基板に依存します。
- **Temperature Sensitivity [mV/ $^{\circ}\text{C}$]**(温度感度): デバイスのデータシートによって提供されるように使われる温度感知器感度。

6. Control Settings(制御設定)

この部分では動的速度要件に対する制御層覆い部(ラッパー)と入力を望む値にすることができます。

- **Loop Control**(繰り返し制御): この任意選択は(速度調整なしで駆動と電動機を同期する)開路制御と(駆動と電動機の同期制御の他に速度制御のためのより遅い制御層が追加される)閉路制御間の切り換えを使用者に許します。
- **Regulator Min Speed [RPM]**(調整器最小速度): 可変抵抗器が0%の時に達成するのを許される最小速度。
- **Regulator Max Speed [RPM]**(調整器最大速度): 可変抵抗器が100%の時に達成するのを許される最大速度。

7. Safety Settings(安全設定)

この部分では障害事象を望む値にして、回復不能な損傷を防ぐのが必要とされる時に電動機を停止することができます。

- **Fault Enable**(障害許可): 障害処理層に対する有効/無効切り換え。この障害任意選択はこの任意選択が禁止される時に隠されます。

- **Fault State**(障害状況): 障害条件中の駆動信号の状況。チェックされていない場合はその入出力(I/O)が0に設定され、チェックされている場合は入出力(I/O)が1に設定されていることを意味します。

▼ Safety Settings

Fault Enable ⓘ

WARNING: Verify H-Bridge Configuration to avoid permanent damage!

Fault State	Pin 0	Pin 1	Pin 2	Pin 3	Pin 4	Pin 5
Output (Low/Hi)	☒	☐	☐	☐	☐	☐

Fault Recovery ⓘ

Stop

Over-Voltage Threshold [V] ⓘ

0 ≤ 50 ≤ 100

Over-Voltage Return [V] ⓘ

0 ≤ 49 ≤ 49

Under-Voltage Threshold [V] ⓘ

0 ≤ 6 ≤ 100

Under-Voltage Return [V] ⓘ

7 ≤ 7 ≤ 100

Over-Heating Threshold [°C] ⓘ

0 ≤ 60 ≤ 150

Over-Heating Return [°C] ⓘ

0 ≤ 50 ≤ 59

Over-Current Peak Threshold [A] ⓘ

0 ≤ 44 ≤ 100

Over-Current Average Threshold [A] ⓘ

0 ≤ 10 ≤ 20

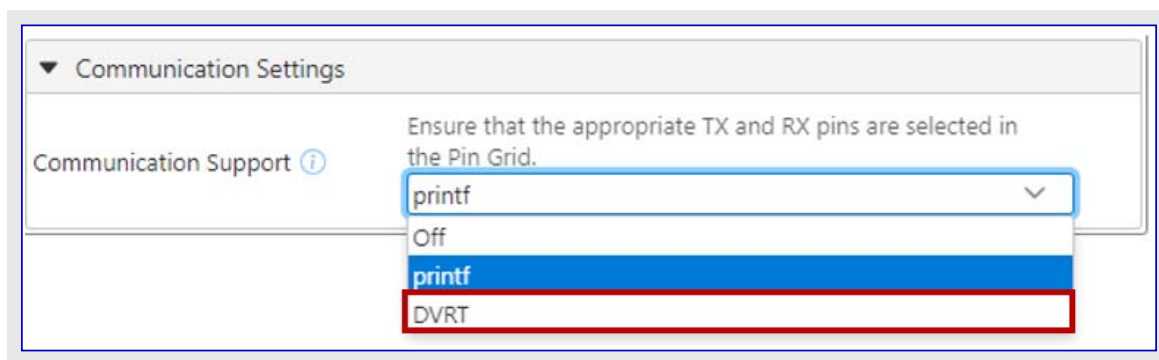
- **Over-Voltage Threshold [V]**(過電圧閾値): Vで与えられる電圧バース最大限度。これはどの回復不能な損傷からもハードウェア構成を守る働きをします。
- **Over-Voltage Return [V]**(過電圧回復): Vで与えられる電圧バース最大回復値。この値に達すると、OVP(過電圧保護)障害が解消されます。
- **Under-Voltage Threshold [V]**(低電圧閾値): Vで与えられる電圧バース最小限度。この値に達しない場合、電動機は開始しません。
- **Under-Voltage Return [V]**(低電圧回復): Vで与えられる電圧バース最小回復値。この値に達すると、UVP(低電圧保護)障害が解消されます。
- **Over-Heating Threshold [°C]**(過熱閾値): 摂氏で与えられる温度限度。これは回復不能な損傷からMOSFETや電動機を守る働きをします。
- **Over-Heating Return [°C]**(過熱回復): 摂氏で与えられる温度回復値。この値に達すると、障害を解消して電動機駆動を再開することができます。
- **Over-Current Peak Threshold [A]**(過電流頂点閾値): Aで与えられる瞬間最大電流限度。これは回復不能な損傷からMOSFETを守る働きをします。
- **Over-Current Average Threshold [A]**(過電流平均閾値): Aで与えられる連続平均電流限度。これは過負荷や固定子と回転子間の位相進みの差が大きすぎる場合に使われます。

8. Communication Settings(通信設定)

この部分では通信形式を望むようにすることができます。このライブラリはシリアル インターフェースとデータ可視器実行時(DVRT:Data Visualizer Run Time)の2つの通信インターフェースを支援します。通信方法のどれかを許可すると、使うプロジェクト資源にUSART単位部が追加されます。

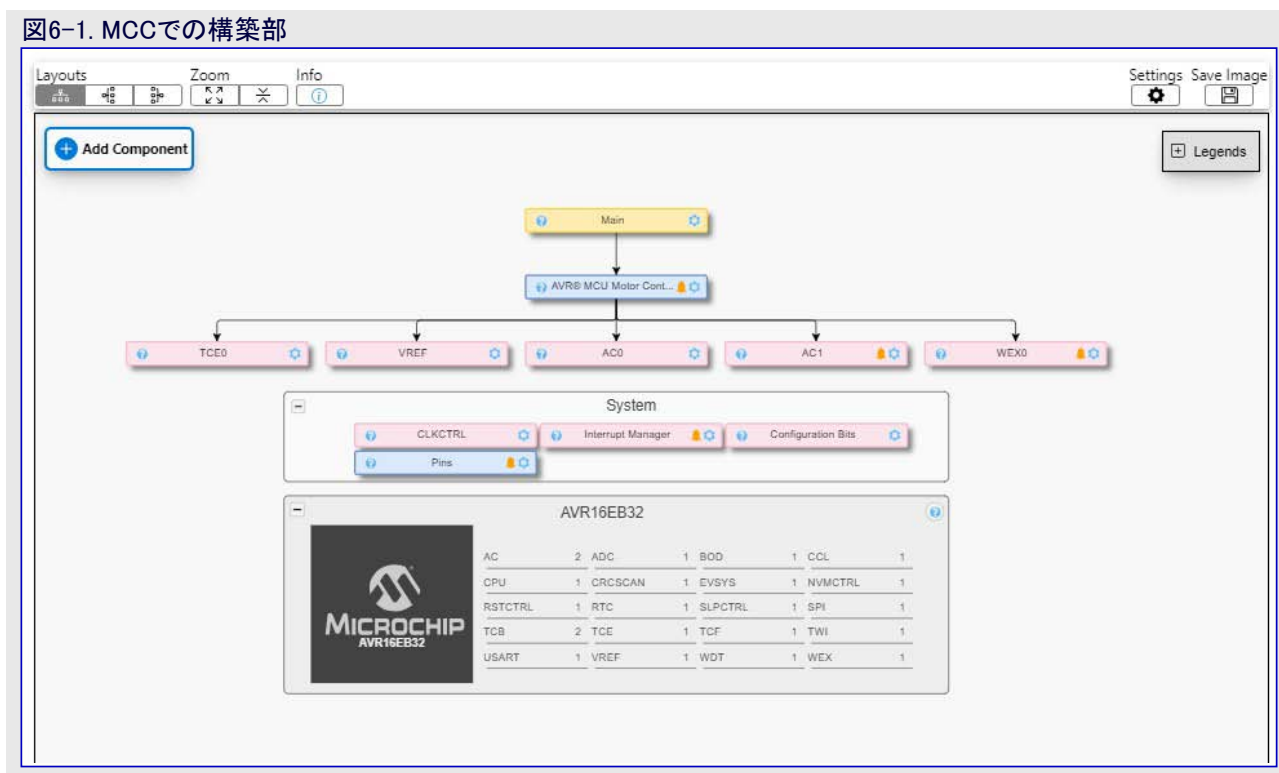
- **Communication Support(通信支援)**⇒**printf**: 既定の460800ボーレートでシリアル インターフェースが許可されます。読み込みはMPLAB データ可視器(Data Visualizer)拡張を含むどれかの道具で行うことができます。端末はRPMでの電動機速度、平均消費電流、入力電圧、温度感知器、%での可変抵抗器値を表示します。

- **Communication Support(通信支援)**⇒**DVRT**: MPLABプラグインのデータ可視器(Data Visualizer)は望む変数を図で実時間作図することができる任意選択を持ちます。生成したDVRT任意選択は可変抵抗器変数と電動機速度変数の描画を許します。



6.2. 応用構築部操作盤

図6-1. MCCでの構築部



6.3. ピン格子操作盤

- BEMF Neutral(逆起電力中点): 電動機のBEMFを検出する比較器入力用中点接続
- BEMF A～C(逆起電力A～C): 電動機のA～C相に対するBEMF採取線
- Hall A～C(ホールA～C): 電動機のA～C相に対するホール感知器採取線
- Current Sense Sum(電流検出合計): 電流検出抵抗測定から増幅した合計電流を提供する信号
- Current Sense Ref(電流検出参照基準): 電流検出増幅器用参照基準入力
- Voltage Sense(電圧検出): 分圧した電圧入力信号
- Temperature(温度): 温度感知器用入力
- Potentionmeter(可変抵抗器): 電動機速度を制御する基板上の可変抵抗器用入力
- Button(釦): 電動機の開始と停止を制御する基板上の釦用入力
- LED(LED): 基板上の表示LED用出力
- UART(UART): シリアル通信線
- WEX0(WEX0): 半ブリッジ駆動部に接続する駆動信号

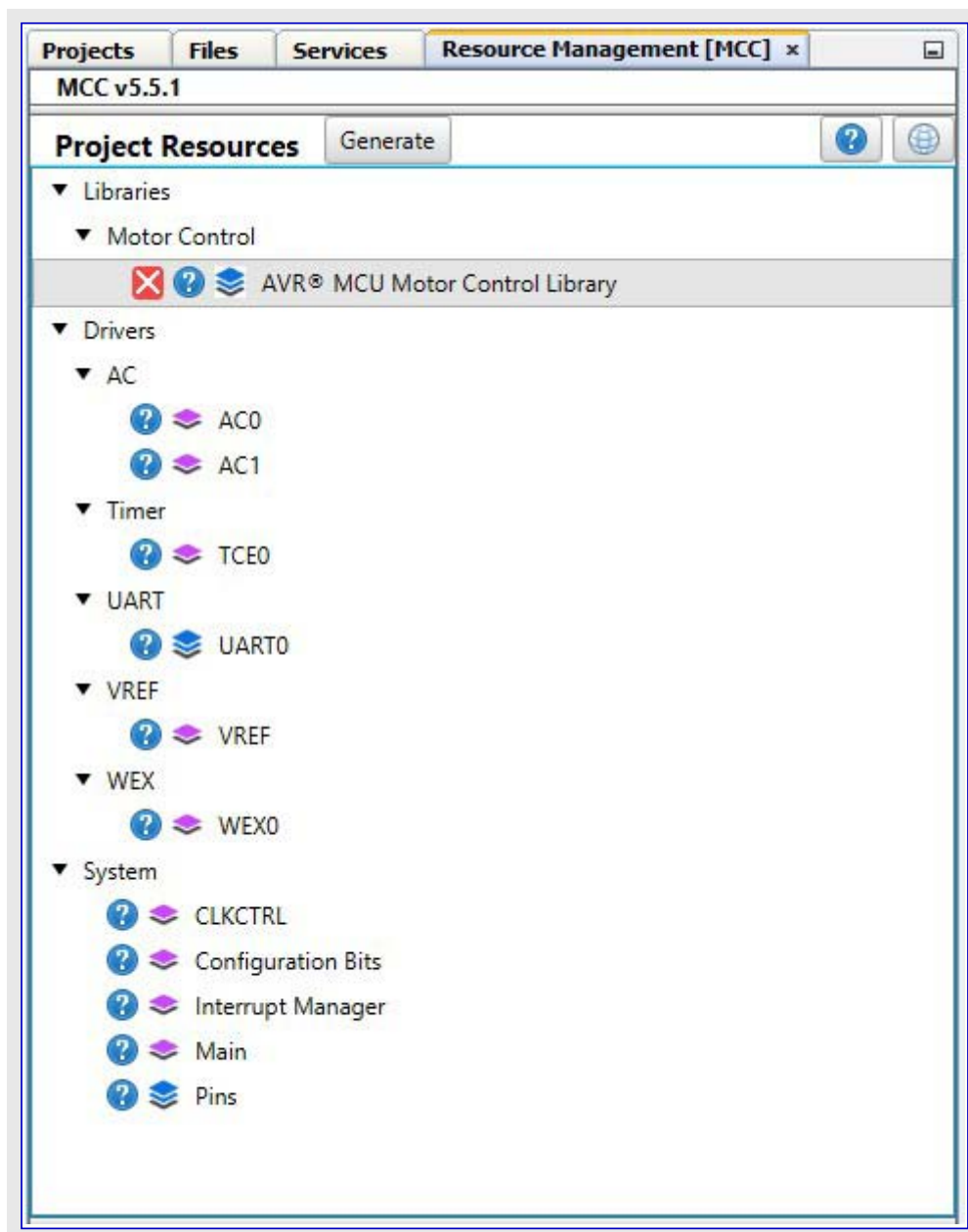
図6-2. BEMFでのピン格子表示

Output			Notifications [MCC]		Pin Grid View x																													
Package:			QFN32 v		Pin No:		30	31	32	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	20	21	22	23	24	25	26	27
							PORTA							PORTC							PORTD							PORTF						
Module	Function	Direction	0	1	2	3	4	5	6	7	0	1	2	3	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7				
AC1	--	--																																
Motor Control v	BEMF Neutral	input																																
	BEMF A	input																																
	BEMF B	input																																
	BEMF C	input																																
	Current Sense Sum	input																																
	Current Sense Ref	input																																
	Voltage Sense	input																																
	Temperature	input																																
	Potentiometer	input																																
	Button	input																																
LED	output																																	
CLKCTRL	CLKOUT	output																																
USART0 v	RXD	input																																
	TXD	output																																
	XCK	output																																
WEX0 v	XDIR	output																																
	WO0	output																																
	WO1	output																																
	WO2	output																																
	WO3	output																																
	WO4	output																																
	WO5	output																																
	WO6	output																																
WO7	output																																	
AC0	--	--																																
Pins	--	--																																

図6-3. ホールセンサーでのピン格子表示

Output			Notifications [MCC]		Pin Grid View ×																													
Package:			QFN32 ▾		Pin No:		30	31	32	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	20	21	22	23	24	25	26	27
							PORTA							PORTC							PORTD							PORTF						
Module	Function	Direction	0	1	2	3	4	5	6	7	0	1	2	3	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7				
AC1 ▶	--	--																																
Motor Control ▼	HALL A	input	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒		
	HALL B	input	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒		
	HALL C	input	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒		
	Current Sense Sum	input			🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒		
	Current Sense Ref	input			🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒		
	Voltage Sense	input			🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒		
	Temperature	input			🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒		
	Potentiometer	input			🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒		
	Button	input	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	
LED	output	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒	🔒		
CLKCTRL ▼	CLKOUT	output								🔒																								
USART0 ▼	RXD	input		🔒		🔒		🔒					🔒					🔒							🔒								🔒	
	TXD	output	🔒		🔒		🔒					🔒								🔒													🔒	
	XCK	output			🔒				🔒					🔒								🔒												
	XDIR	output				🔒				🔒													🔒											
WEX0 ▼	WO0	output	🔒		🔒						🔒							🔒								🔒								
	WO1	output		🔒		🔒						🔒							🔒								🔒							
	WO2	output			🔒		🔒					🔒								🔒								🔒						
	WO3	output				🔒		🔒					🔒								🔒								🔒					
	WO4	output					🔒		🔒					🔒								🔒								🔒				
	WO5	output						🔒		🔒					🔒								🔒								🔒			
	WO6	output							🔒							🔒								🔒							🔒			
	WO7	output								🔒															🔒									
AC0 ▶	--	--																																
Pins ▶	--	--																																

6.4. 資源管理操作盤



7. 生成したAPI

生成したライブラリは低位構成設定を抽象化して簡単にした使い易いAPIの一覧を提供します。

`motor_control.h`で見つかる電動機制御階層からのAPIは以下です。

- `MC_Initialize` – この初期化関数は他のどの関数よりも先に呼ばなければなりません。
- `MC_StartStop` – 開始/停止の事象が受け取られた時に見つかる状態に応じて電動機を開始または停止します。
- `MC_DelayMs` – 割り込みなしで背面で計時器を使って遅延を実行します。
- `MC_ReferenceSet` – 閉路での速度に対する参照基準点または開路同期での振幅基準を設定します。
- `MC_PotentionmeterRead` – %で表現される可変抵抗器値を返します。
- `MC_FastPotentionmeterRead` – 生のアナログ/デジタル(A/D)形式で表現される可変抵抗器値を返します。
- `MC_VoltageBusRead` – Vで表現される電圧バス値を返します。
- `MC_TemperatureRead` – 摂氏で表現されるMOSFETトランジスタ温度を返します。
- `MC_CurrentRead` – mAで表現される平均応用消費電流を返します。
- `MC_Speed_Get` – `mc_speed_t`で表現される回転速度を返します。
- `MC_PeriodicHandlerRegister` – 主応用に対して独自ソフトウェア呼び戻しを登録します。この関数は割り込み状況ではなく、`main`から呼ばなければなりません。
- `MC_StatusGet` – 電動機の状態(`MOTOR_IDLE`、`MOTOR_RUNNING`、または`MOTOR_FAULT`)、電動機の方法(`MC_DIR_CW`または`MC_DIR_CCW`)、障害状態(`FAULT_UNDERVOLTAGE`、`FAULT_OVERVOLTAGE`、`FAULT_STALL`、`FAULT_OVERTEMPERATURE`、または`FAULT_OVERCURRENT`)を返します。

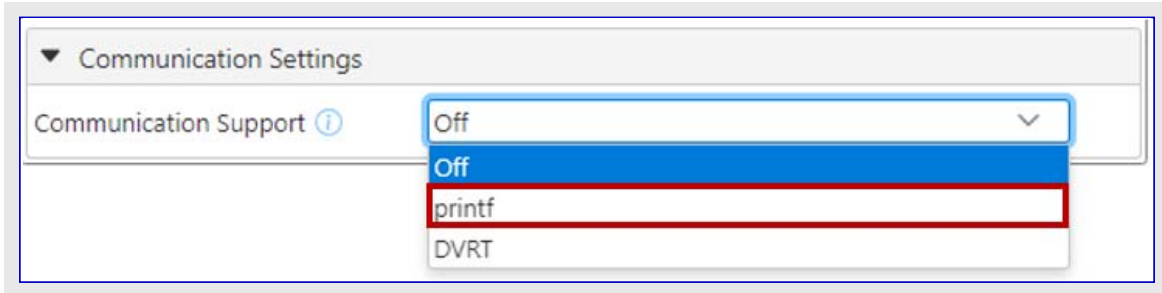
使用者はライブラリの機能を変えないように`motor_control.h`ファイルの公開APIだけを使わなければなりません。

8. 通信

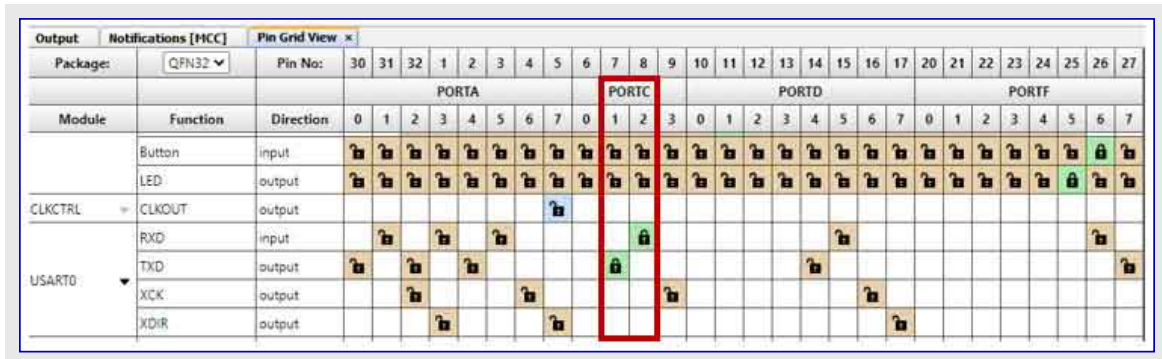
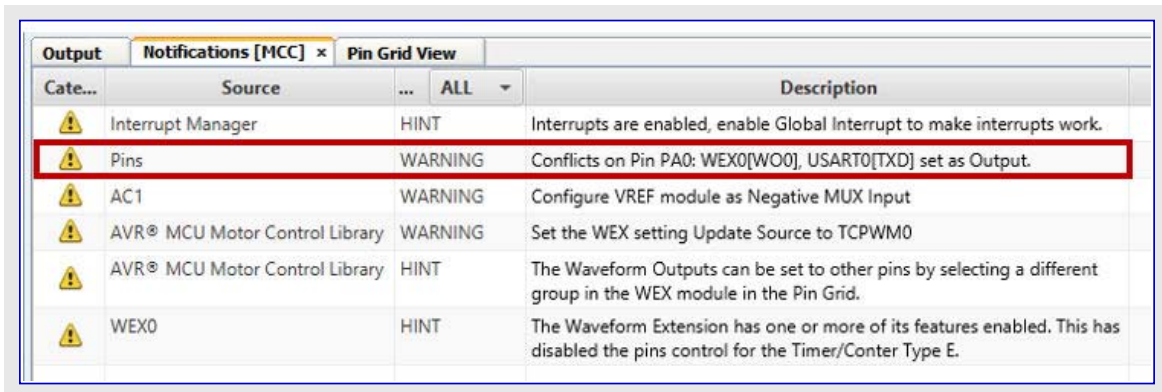
以下の手順は多相電力基板キットを使って支援した動作のどちらかで基板との通信の確立を示します。

8.1. 端末経由の通信 (printf)

ライブラリのCommunication Settings(通信設定)部分で”**printf**”支援を選んでください。



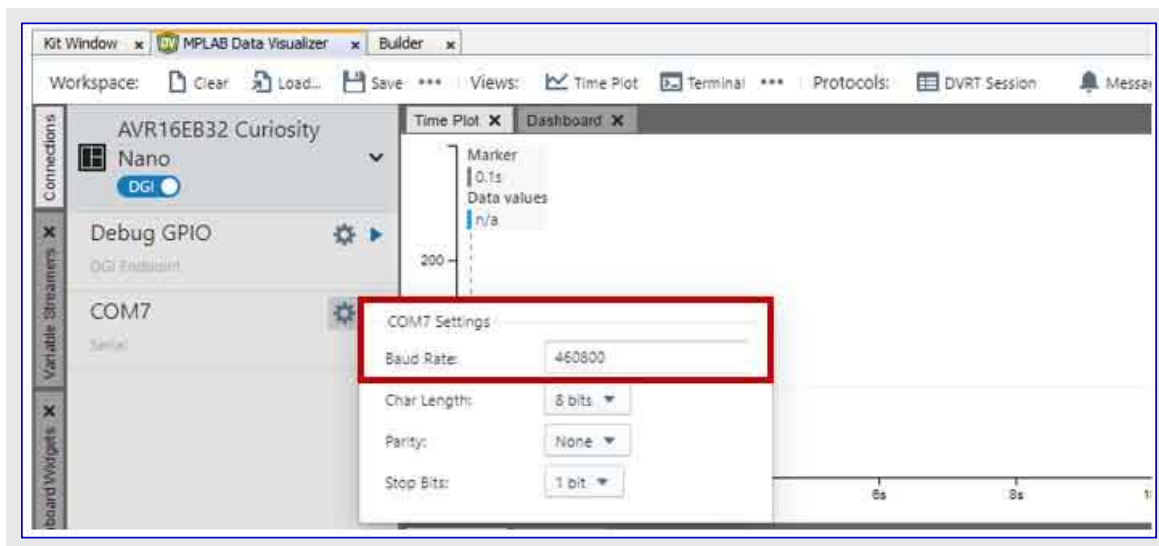
ピン競合の警告が現れる場合、Pin Grid View(ピン格子表示)タブに切り換えてRXDとTXDに対してPORTCピンを選んでください。この文書の最初で記述したものの他に別の設定を使う場合、望むポートピン構成設定を選んでください。



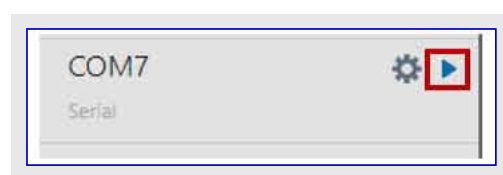
生成してプロジェクトを構築し、デバイスに書いてください。MPLABツールバーでデータ可視器(Data Visualizer)を開いてください。



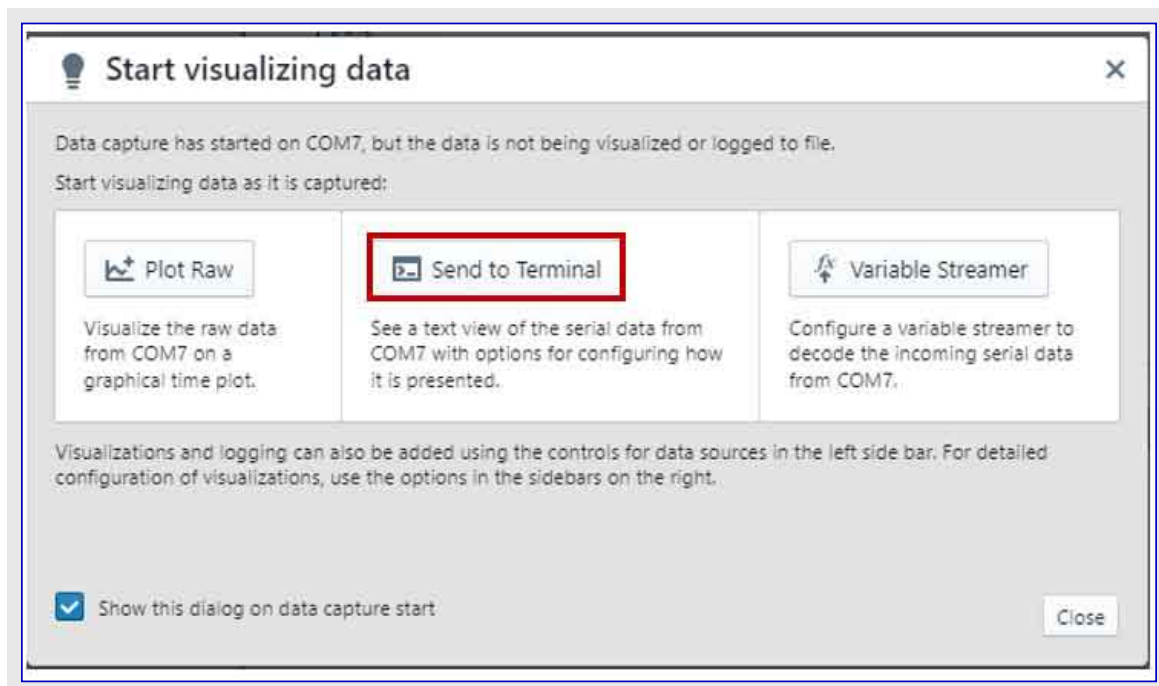
MPLAB Data Visualizer(データ可視器)タブでCuriosity NanoのCOMチャネル設定を選んでBaud Rate(ボーレート)を460800に更新してください。



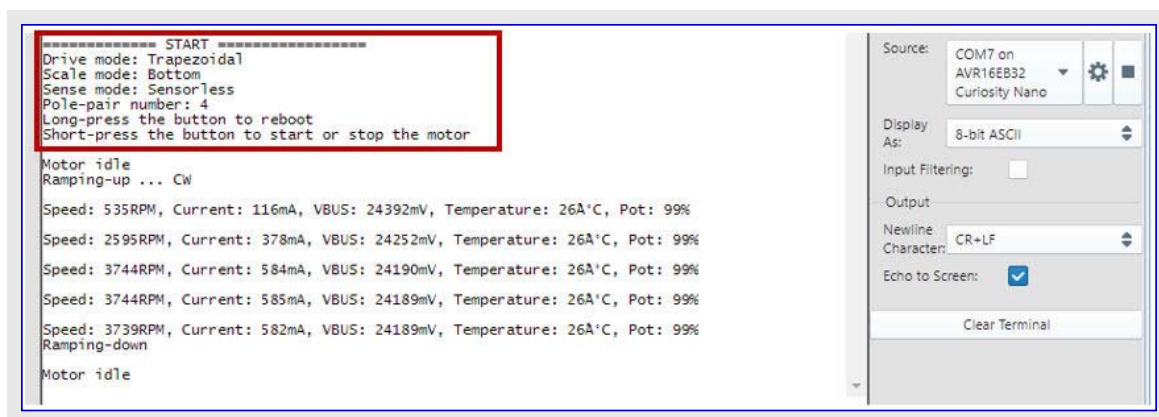
通信を開始するためにConnect(接続)鉤を選んでもください。



促された時にSend to Terminal(端末へ送る)任意選択を選んでください。



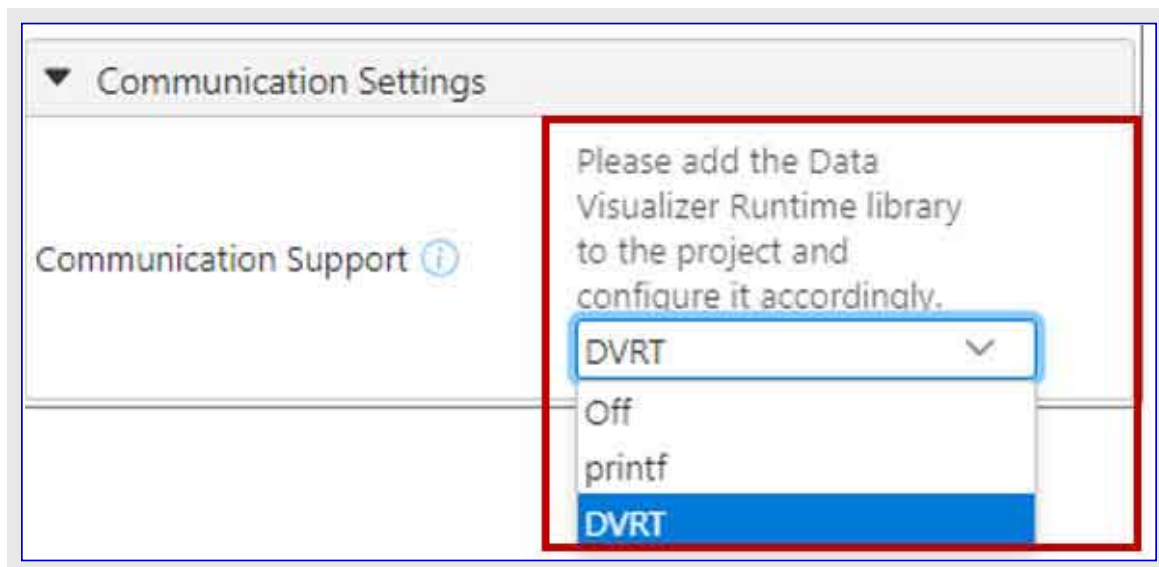
COMポートは通信線が有効の時に状態を”CAPTURING”に変えます。再起動するために多相電力基板(MPPB)の鉤を長押しして端末での開始メッセージ指示をご覧ください。



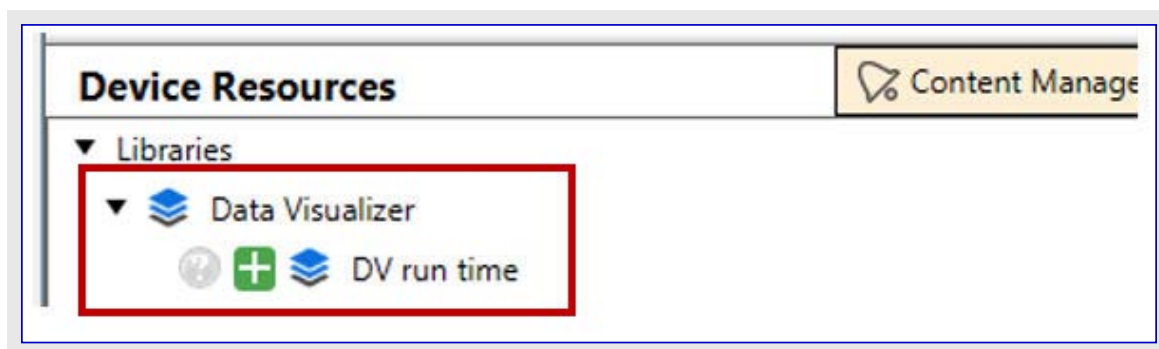
短く押した後、電動機が加速し、端末がRPMでの電動機速度、平均消費電流、入力電圧、温度感知器、%での可変抵抗器値を示します。

8.2. DVRT経由の通信

Motor Control Library(電動機制御ライブラリ)⇒Communication Support(通信支援)でDART任意選択を選んでください。選択引き落としアイコンの上で提供される指示に従ってください。



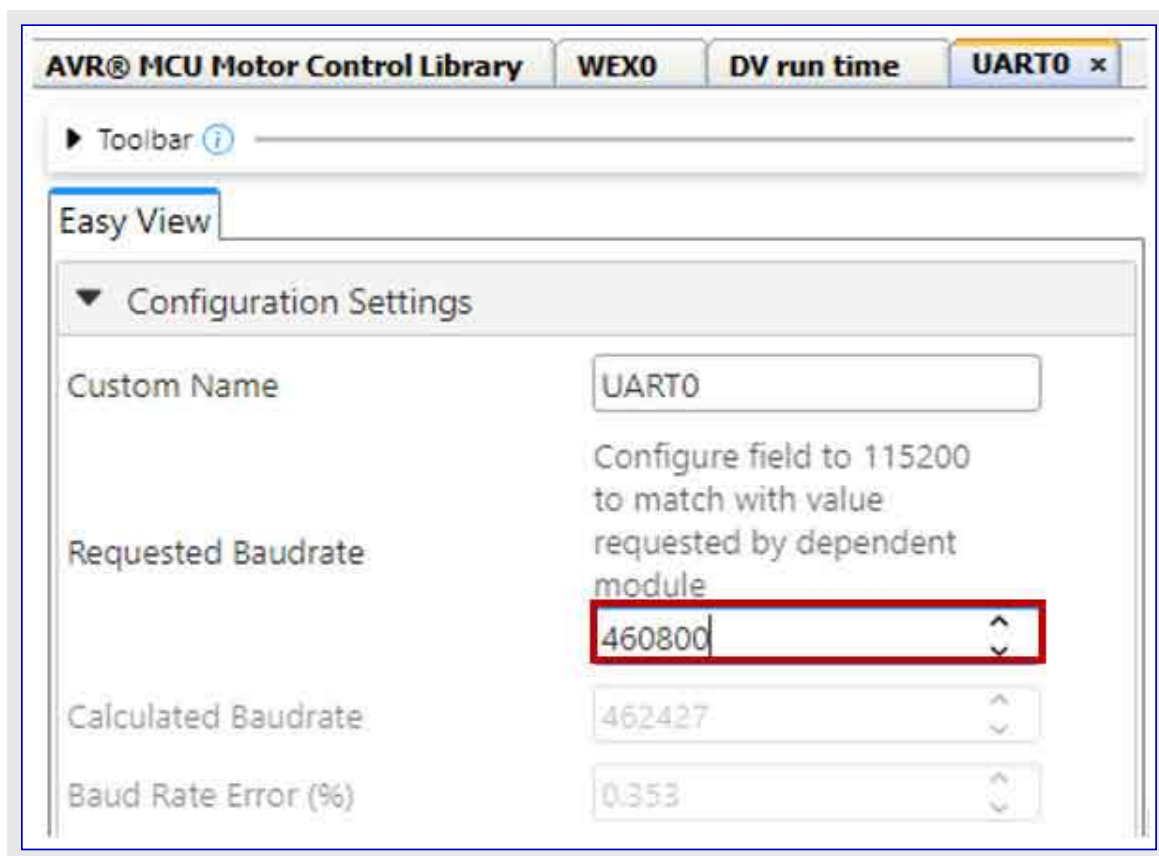
Device Resources(デバイス資源)⇒Libraries(ライブラリ)⇒Data Visualizer(データ可視器)でDV run time(データ可視器走行時)を追加してください。



Project Resources(プロジェクト資源)でUART0を選んでください。



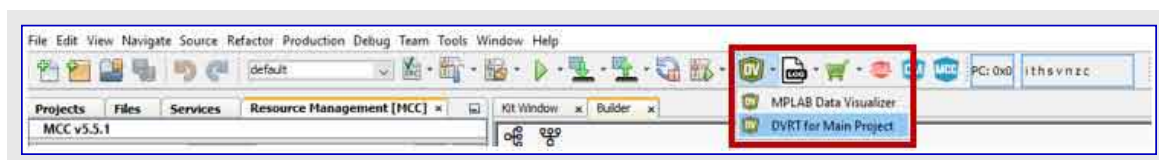
Requested Baudrate(要求されたボーレート)を460800に更新してください。



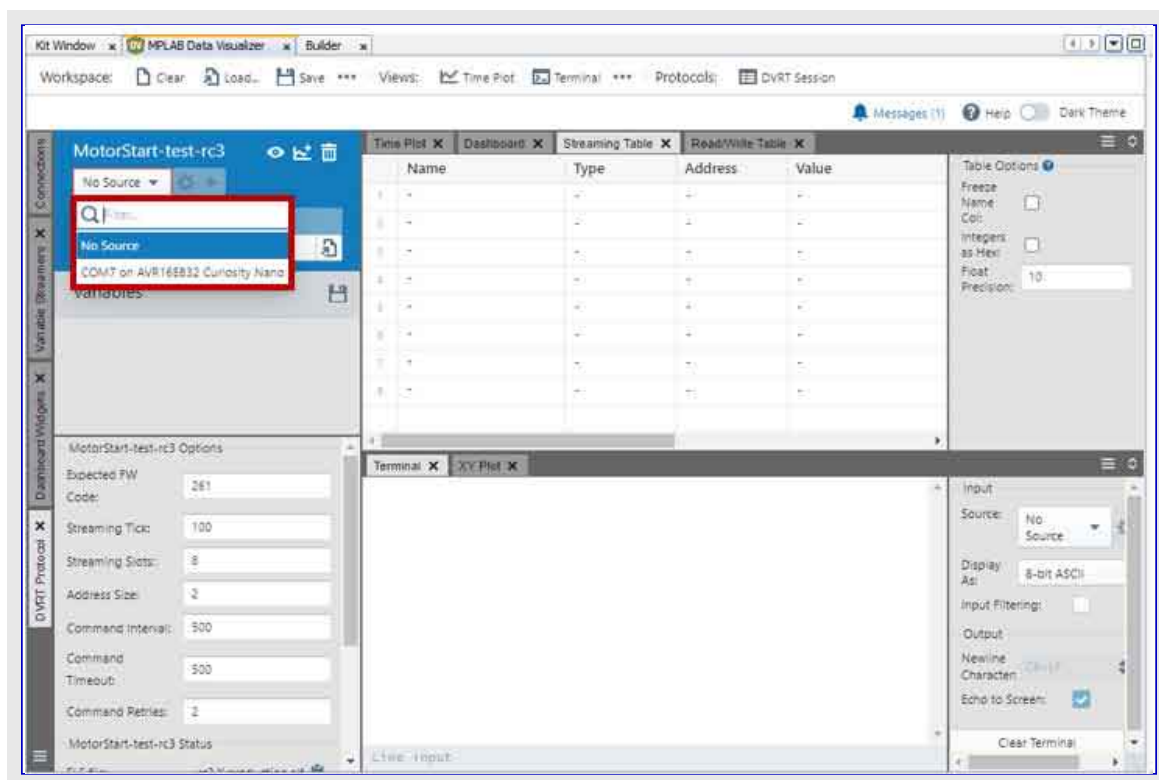
RXDとTXDに対してPORTCピン接続を選んでください。



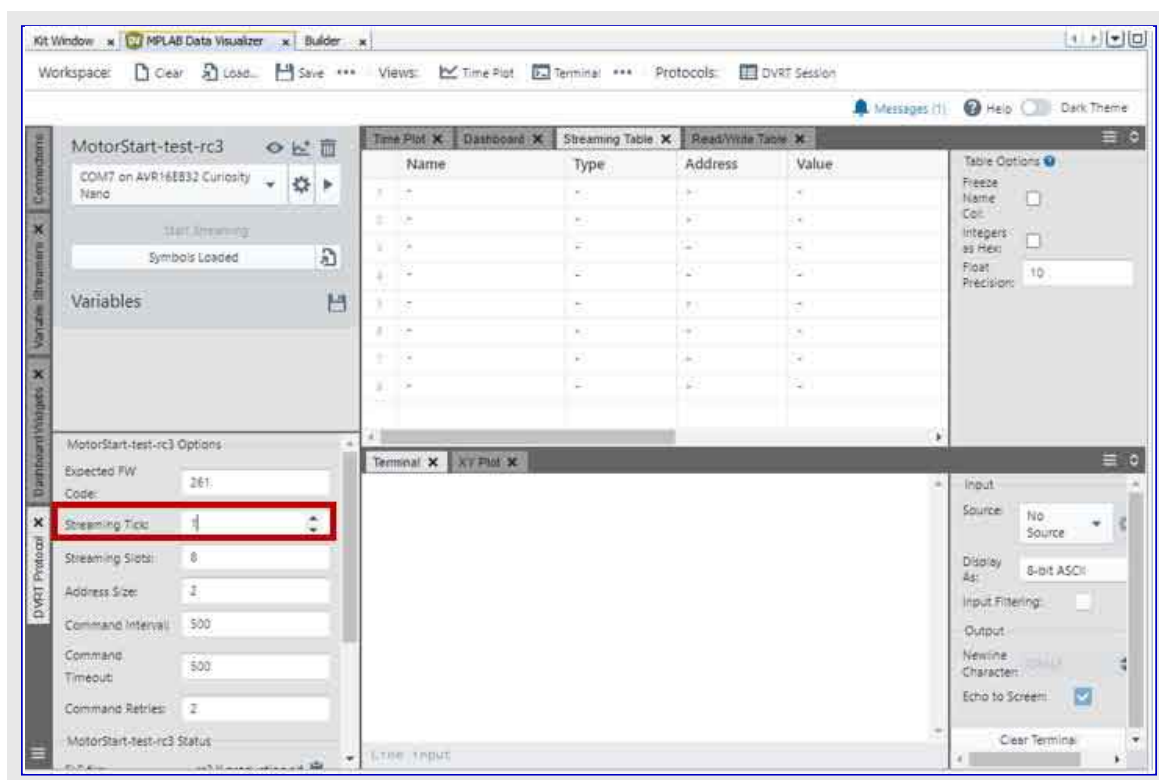
生成、完全構築、書き込みをしてDVRT for Main Project(主プロジェクト用DVRT)を開いてください。



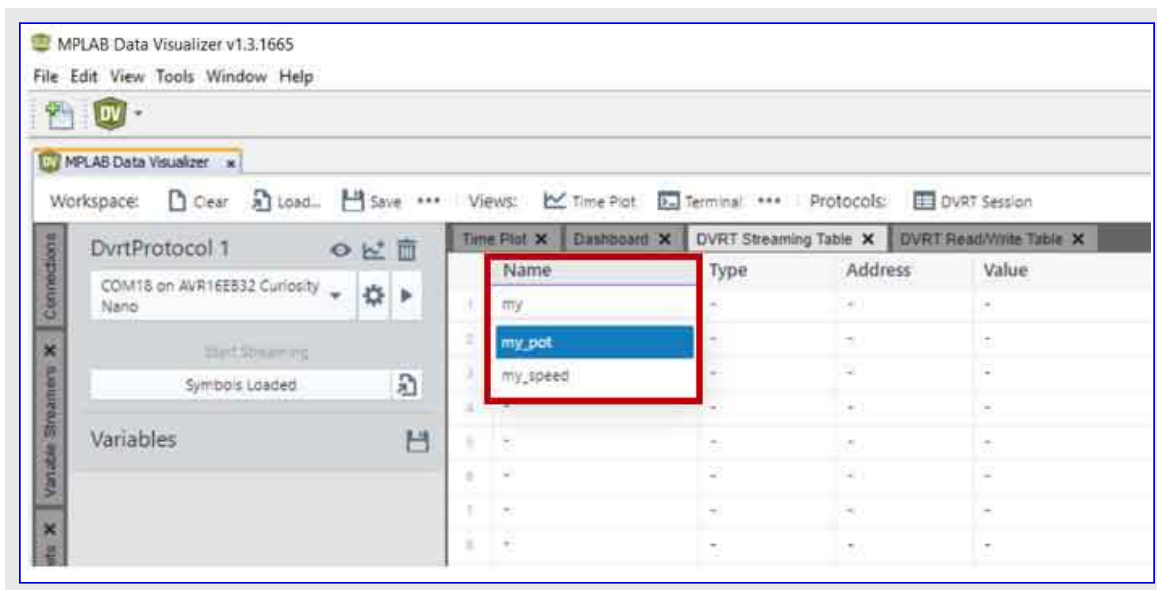
COMn on AVR16EB32 Curiosity Nano接続を選んでください。



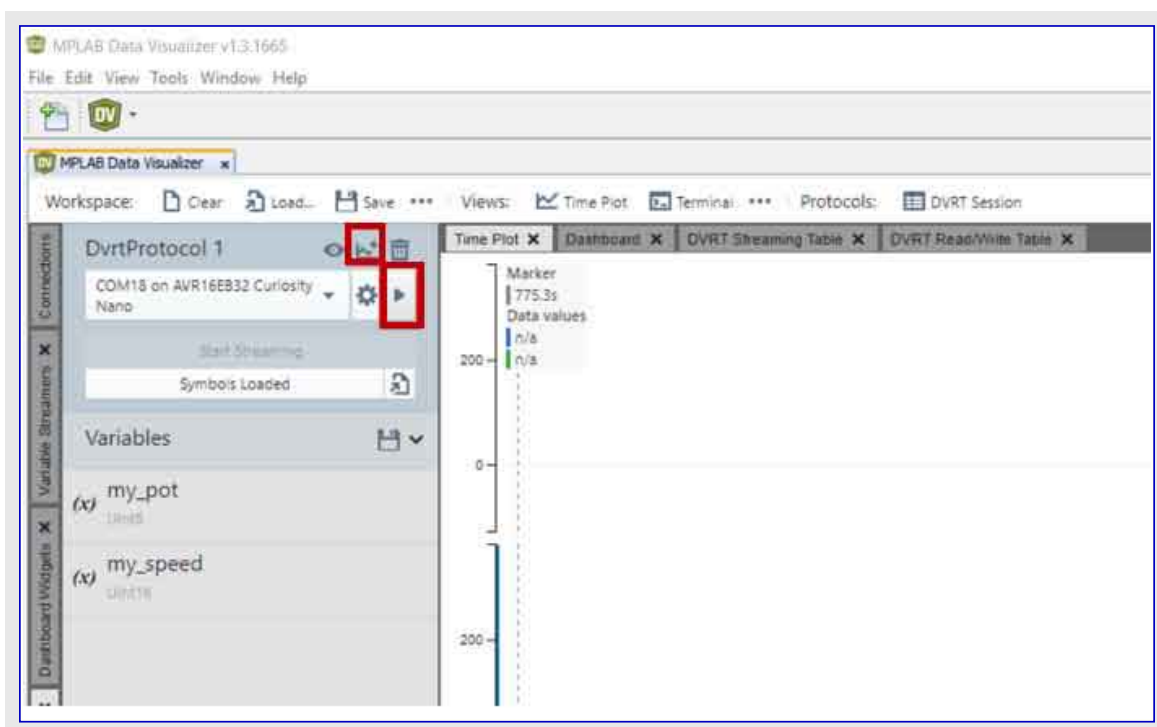
Streaming Ticks(流し刻み)を1に変更してください。



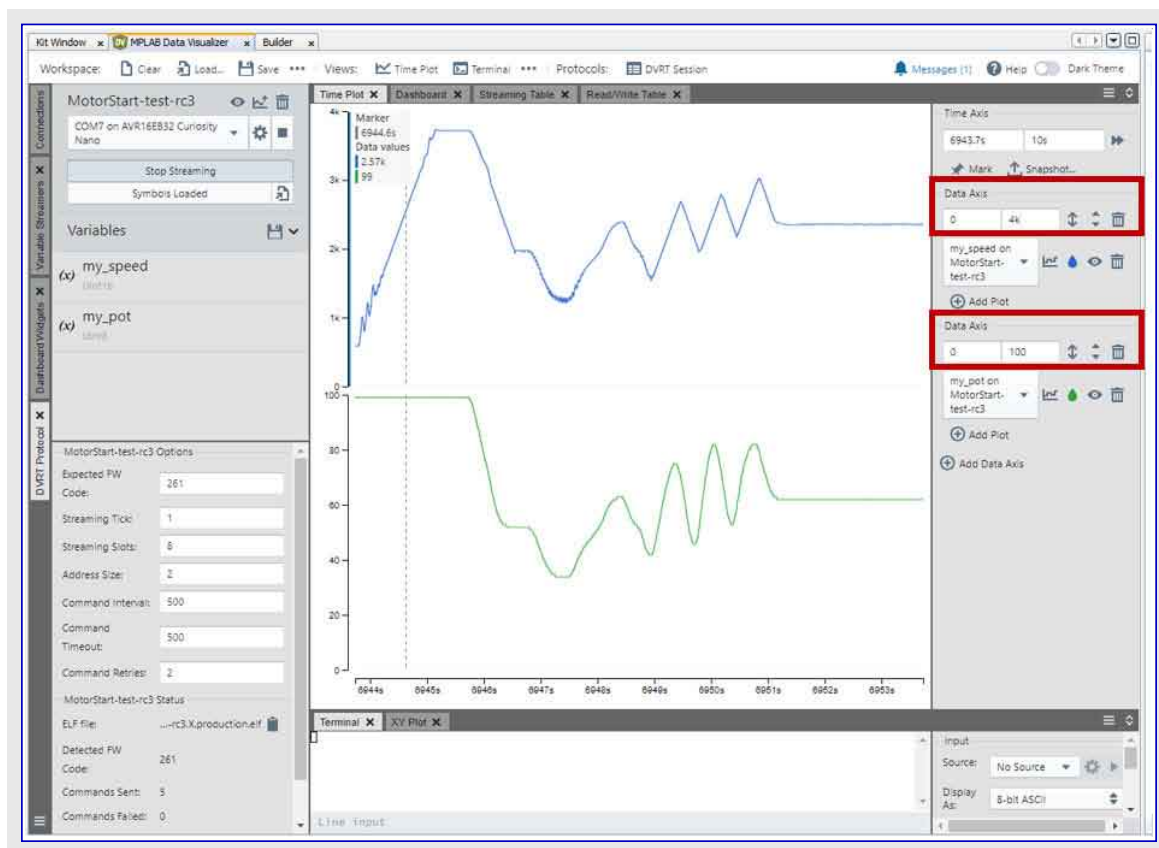
speed(速度)とpot(可変抵抗)の変数を追加してください。



Play(活動)鉤とPlot(全変数作図)鉤をクリックしてください。



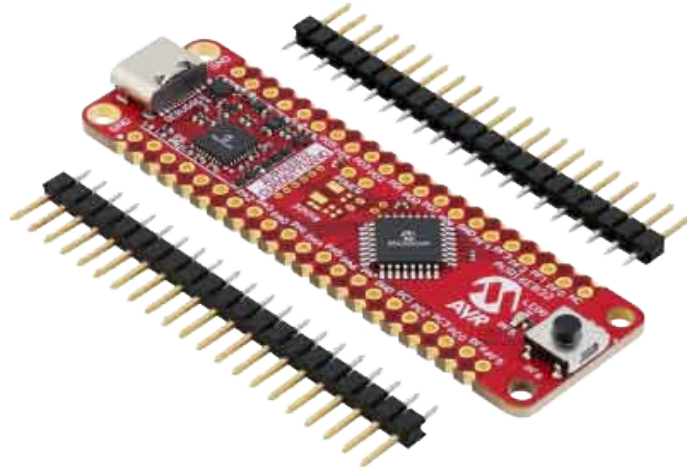
Data Axis(データ軸)を調整してください。my_speedでは0～電動機の公称RPM速度、my_potでは0～100を設定してください。可変抵抗器値を変えている間の結果をご覧ください。



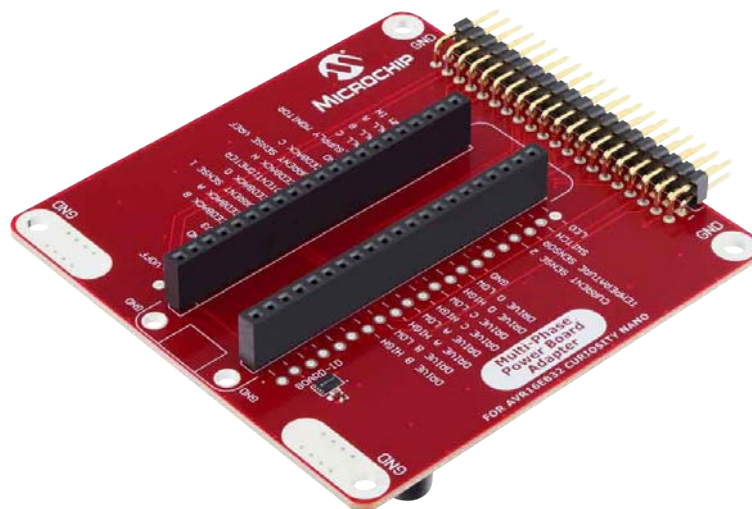
9. 実演

本章では多相電力基板キットを使って台形感知器なしをプログラミングするための完全な段階的な手引きが記述されます。ここはこの実演を再構築するのに必要とされるハードウェアの一覧です。

- AVR16EB32 CNANO – [EV73J36A](#)
 - AVR16EB32 CNANO
 - USB C型ケーブル×1



- AVR16EB32 CNANOの多相電力基板アダプター – [EV88N31A](#)



- 多相電力基板(MPPB) – [EV35Z86A](#)

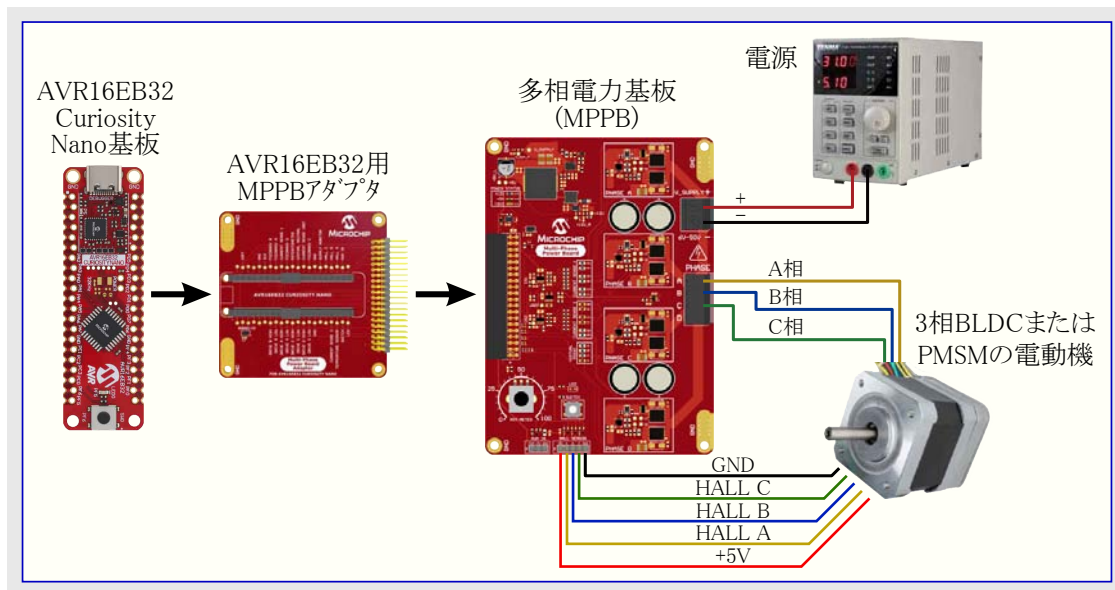


- 3相BLDCまたはPMSMの電動機
- 調整可能な万能電源または最低24V/3Aの固定電源

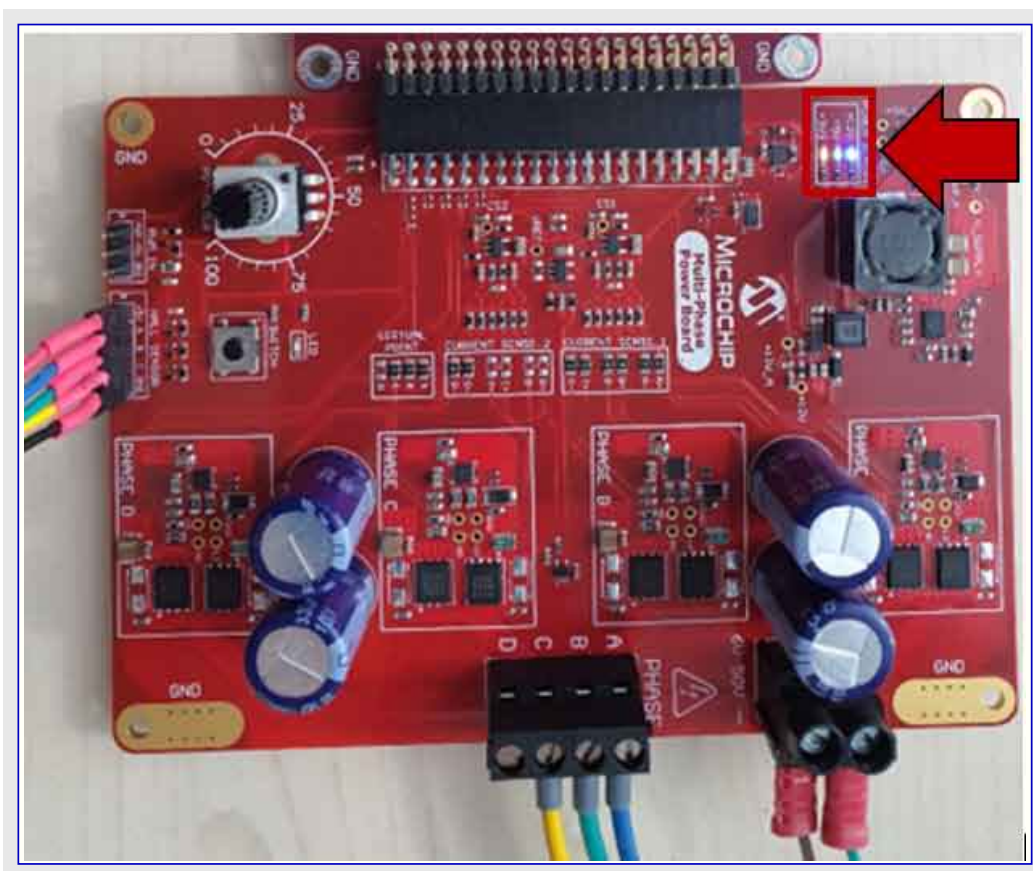
10. 手順

10.1. 手順1. ハードウェア設定

1. MPPBアダプタと多相電力基板と共にCNANO基板を組み立ててください。



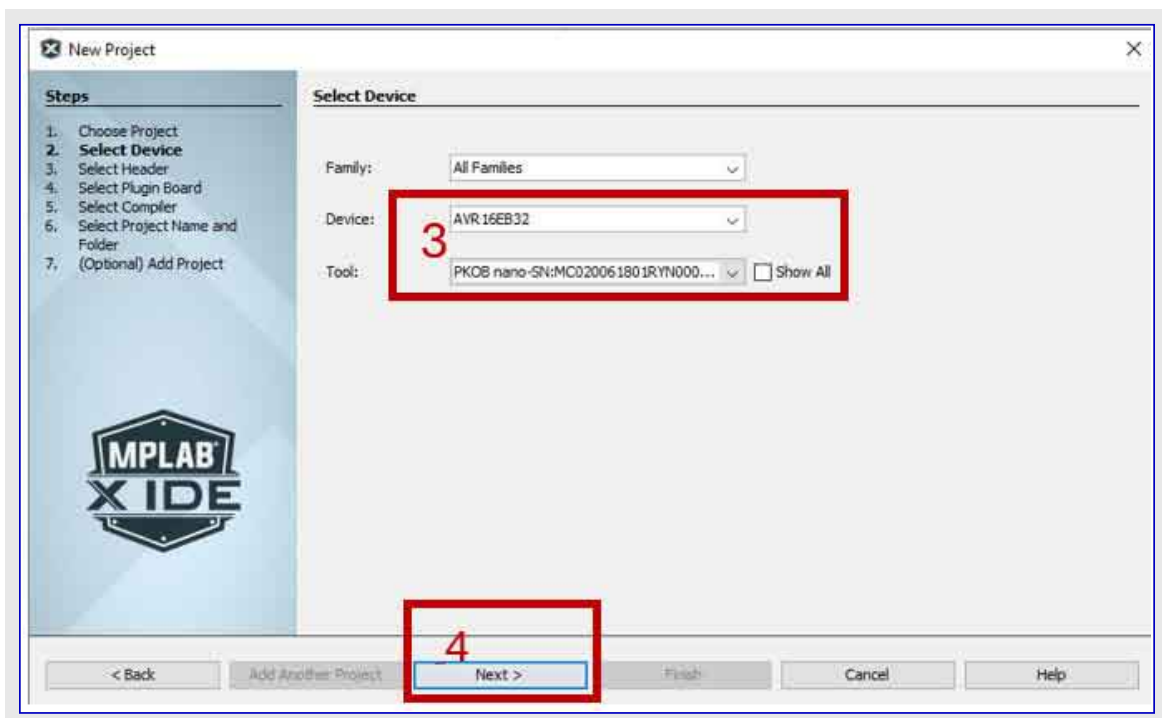
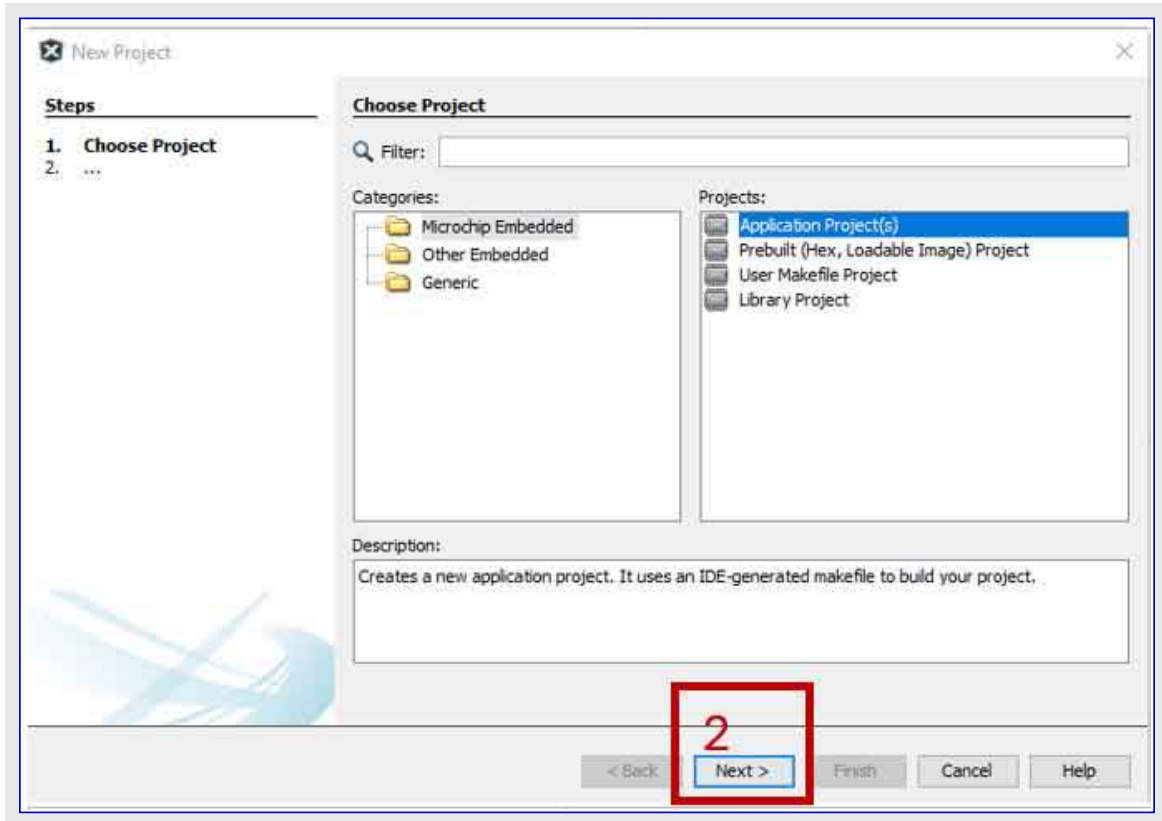
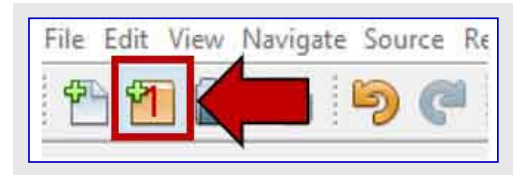
2. CNANO基板をUSB-Cケーブルの手助けでPCに接続してください。
3. 電動機をMPPBに接続してください。ホール感知器使用時、ホール感知器は電動機のデータシートで記述されるように、電動機相と同じ順になるのを確実にしてください。
4. 電源を接続して電圧を電動機の公称電圧(24V)に設定してください。12V LEDがONにならないとなりません。電力が正しく接続されると、MPPBのLEDは必要とするDC電圧水準を示します。

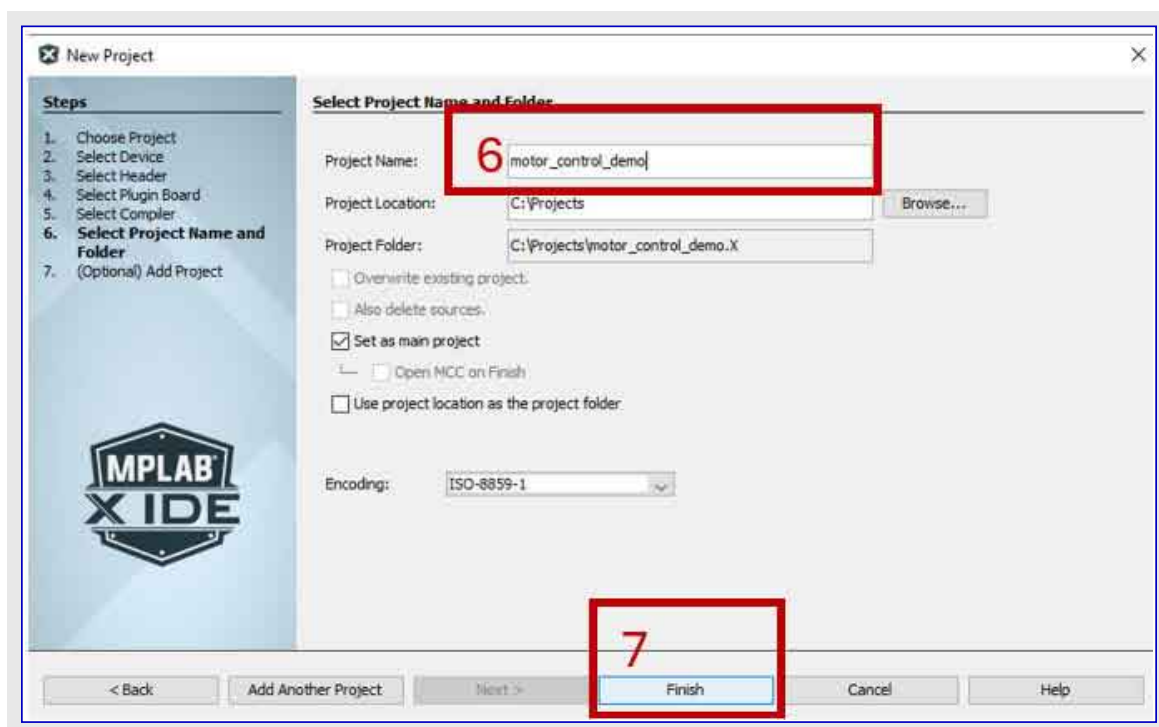
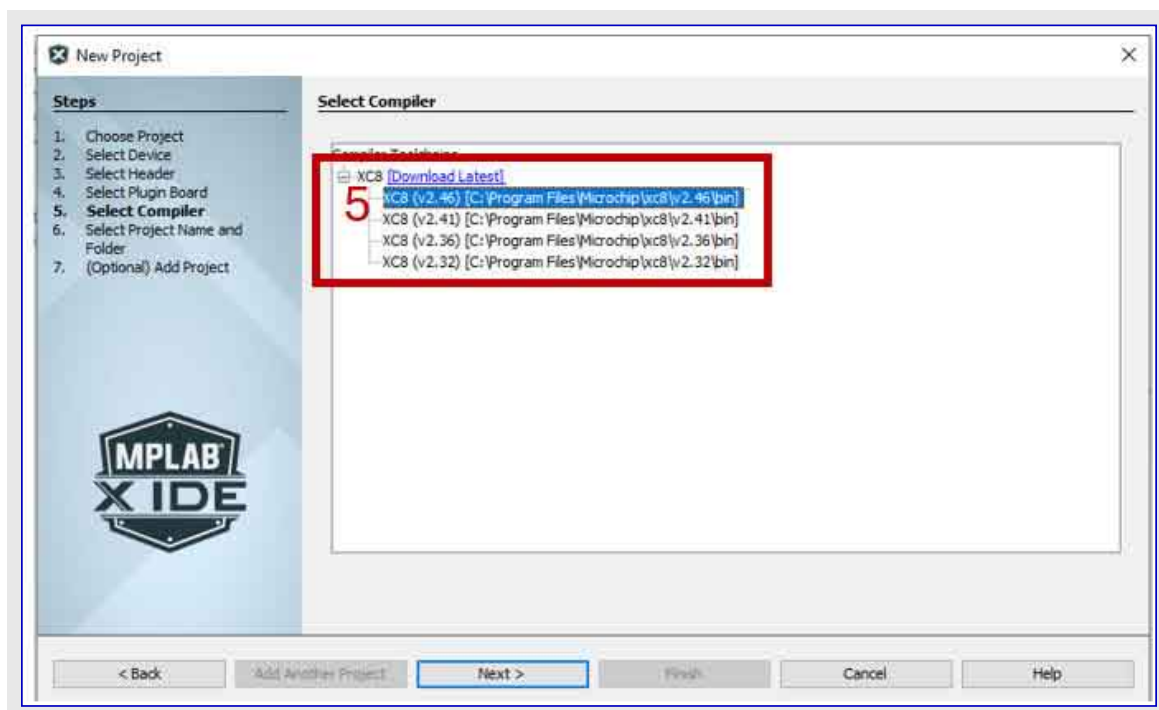


ハードウェアの準備は終わりました。

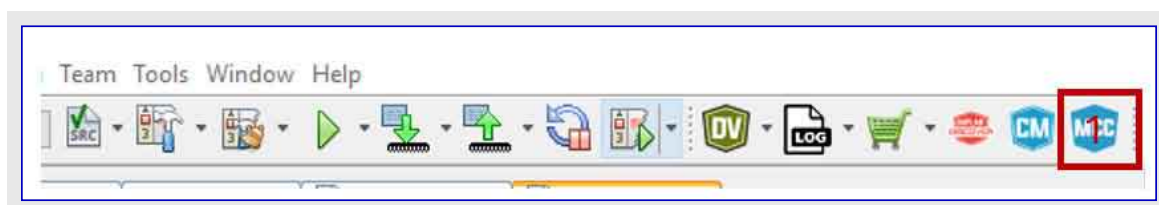
10.2. 手順2. MPLAB®設定

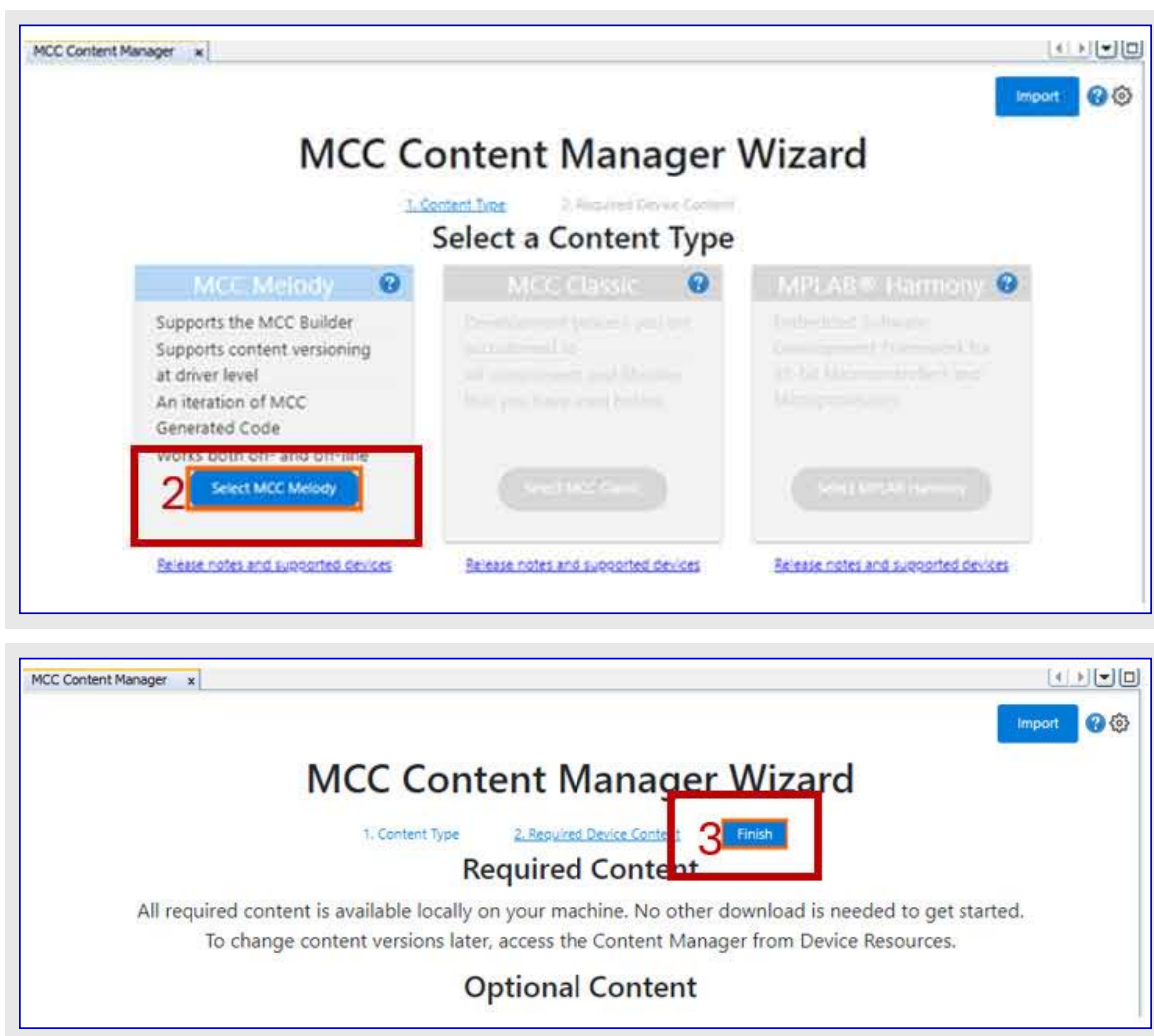
1. MPLAB X IDEを開いてAVR16EB32デバイス用の新規プロジェクトを作成してください。



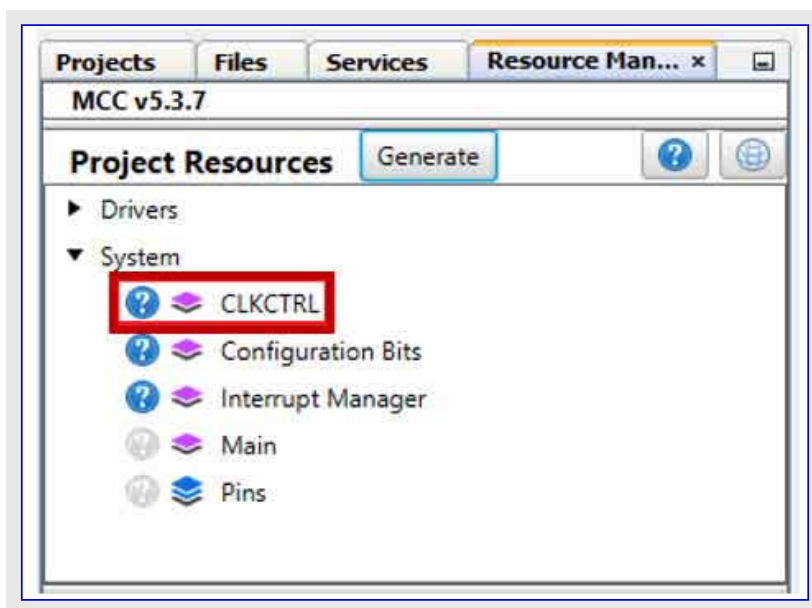


2. ツールバーからMCCを開いてください。ここでMCCプラグインをインストールするより多くの情報が見つかります。





3. Resource Manager(資源管理部)タブでProject Resources(プロジェクト資源)⇒System(システム)⇒CLKCTRL(クロック制御器)をクリックしてください。CLKCTRL(クロック制御器)タブでPrescaler enable(前置分周器許可)切替器を禁止(20MHzコア クロック許可)にしてください。



CLKCTRL x

Easy View

Register Initialization

▼ CLKCTRL Settings

Main Clock (Hz)

20000000

Clock select

Internal high-frequency oscillator

?

External Clock (Hz)

3000000

Prescaler enable

☐

System Clock Out Enable

☐

?

Timebase

20

?

Prescaler division

Divide by 6

?

Prescaler B division

No division

?

▼ PLL Settings

PLL Multiplication Factor

PLL Disabled

?

PLL Source

OSCHF

?

PLL Source Division

DIV1

?

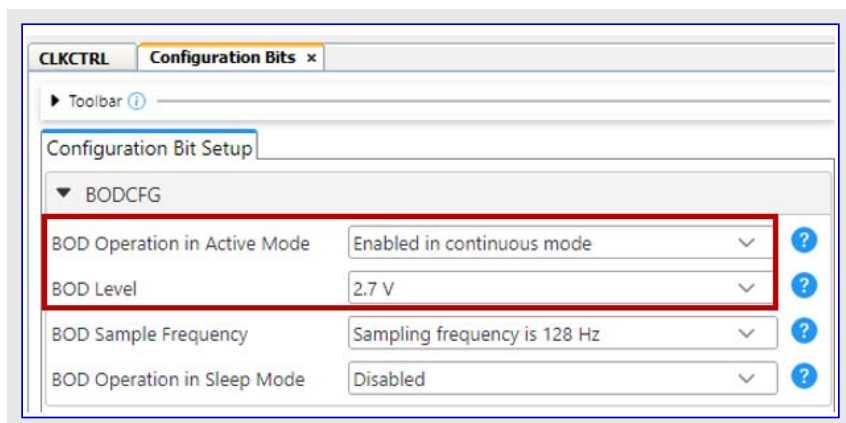
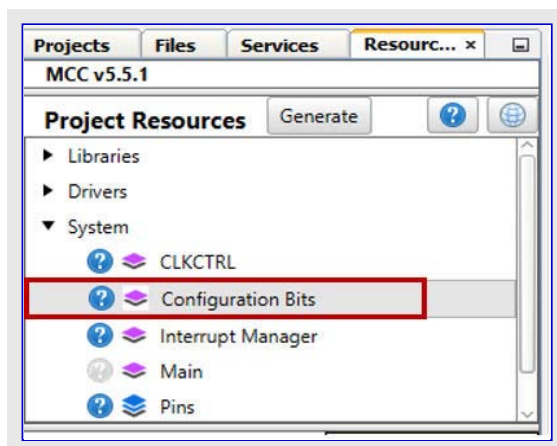
PLL Output Clock Division

NONE

?

▼ Advanced Settings

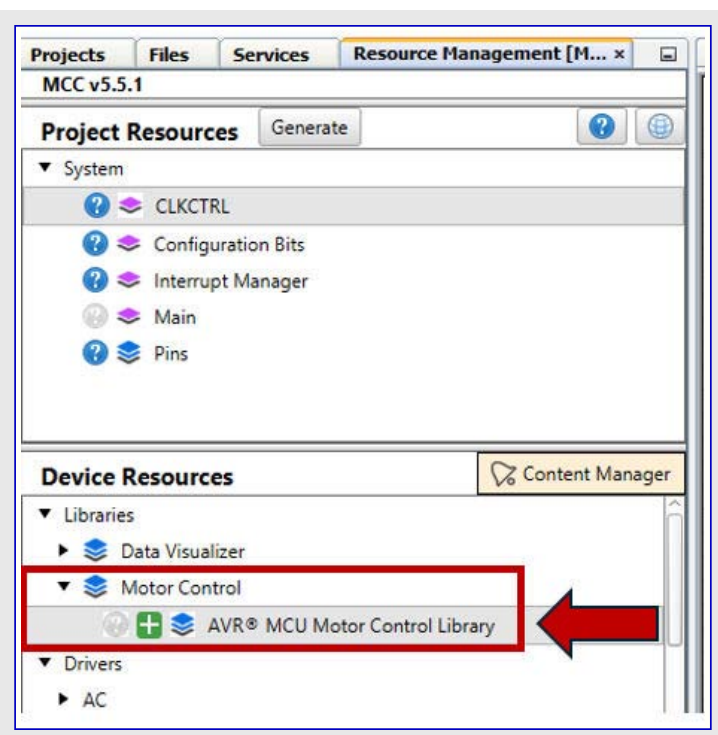
4. BOD設定。System Resource(システム資源)でConfiguration Bits(構成設定ビット)を選んでください。開いたタブでBOD Operation in Active Mode(活動動作でのBOD動作)を”Enabled in continuous mode(継続動作で許可)”とBOD Level(BOD基準)を”2.7V”に変更してください。この段階は電力がOFFにされた時の正しいMCUリセットを許します。



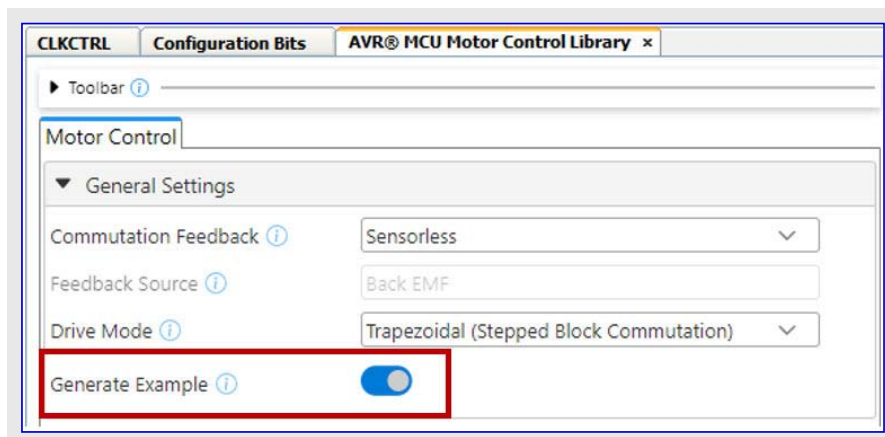
20MHzクロックに設定したMCU設定でMPLAB設定は終わりました。

10.3. 手順3. AVR® MCC電動機制御ライブラリ設定

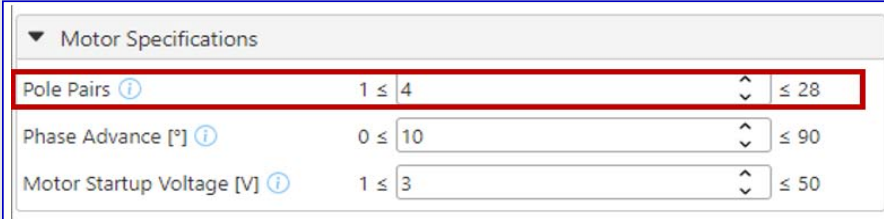
1. 電動機制御ライブラリを追加するため、Device Resources(デバイス資源)⇒Libraries(ライブラリ)⇒Motor Control(電動機制御)⇒AVR MCU Motor Control Library(AVR MCU電動機制御ライブラリ)へ行き、緑の+記号をクリックしてください。



2. Motor Control Library Configuration(電動機制御ライブラリ構成設定)タブで電動機制御層を含める他にすぐに使える実演層も作成するためにGenerate Example(例生成)任意選択を許可してください。これは既定MPPB関連設定とピン接続を仕上げます。

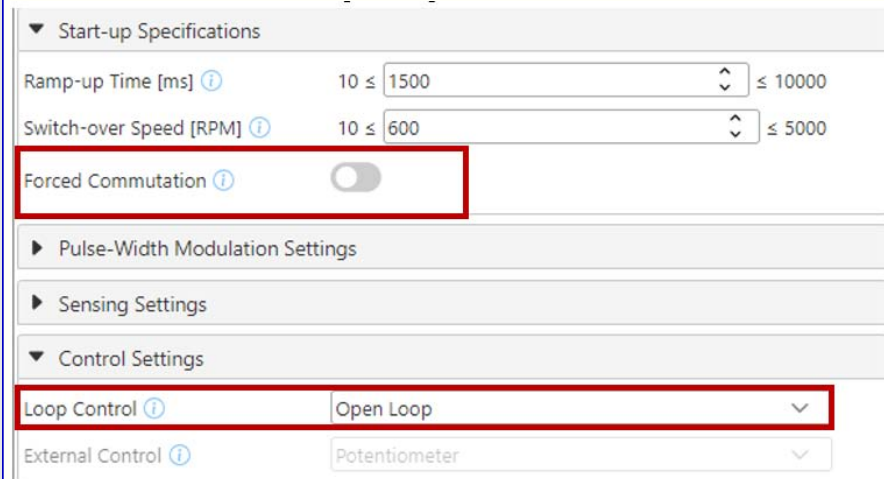


3. 電動機のデータシートを見て極対数と公称電圧を見つけてください。正しい極対を設定してください(実演に使った8極を持つACT57 BLF02の場合、従って4極対)。Phase Advance [°](位相進み)とMotor Startup Voltage [V](電動機始動電圧)は変わらずそのままです。



▼ Motor Specifications		
Pole Pairs ⓘ	1 ≤ 4	≤ 28
Phase Advance [°] ⓘ	0 ≤ 10	≤ 90
Motor Startup Voltage [V] ⓘ	1 ≤ 3	≤ 50

4. Forced Commutation(強制整流(転流))をOFFにしてLoop Control(繰り返し制御)でOpen Loop(開路)を選んでください。



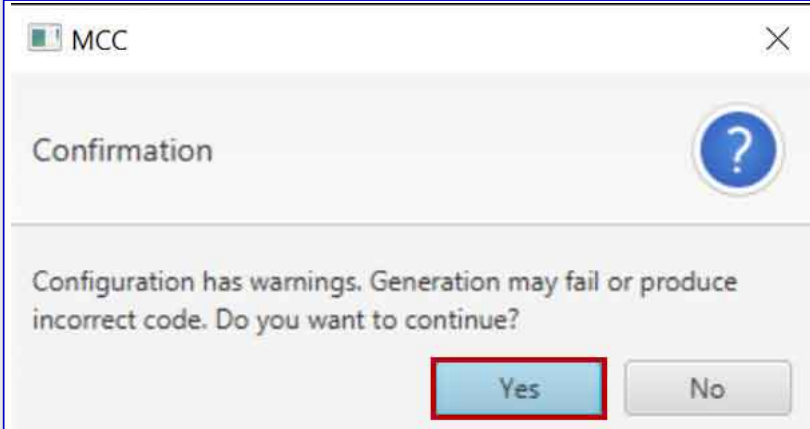
▼ Start-up Specifications	
Ramp-up Time [ms] ⓘ	10 ≤ 1500 ≤ 10000
Switch-over Speed [RPM] ⓘ	10 ≤ 600 ≤ 5000
Forced Commutation ⓘ	☐
▶ Pulse-Width Modulation Settings	
▶ Sensing Settings	
▼ Control Settings	
Loop Control ⓘ	Open Loop
External Control ⓘ	Potentiometer

5. APIと電動機制御ライブラリとで実演コードを生成してください。



Projects	Files	Services	Resourc... x
MCC v5.5.1			
Project Resources		Generate	?

6. 警告付き生成を承認するためにYes(はい)をクリックしてください。



MCC

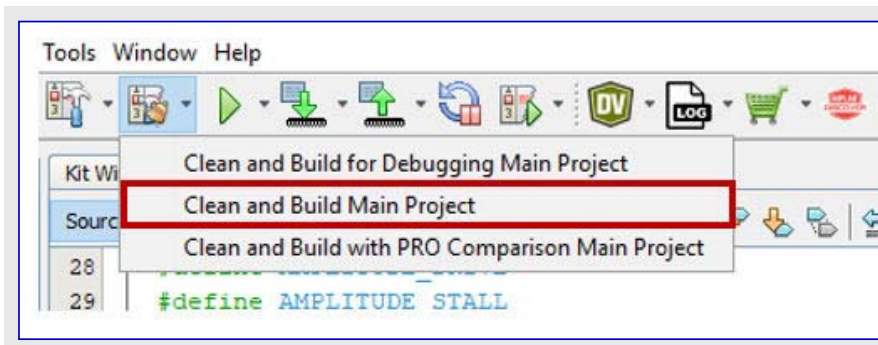
Confirmation

Configuration has warnings. Generation may fail or produce incorrect code. Do you want to continue?

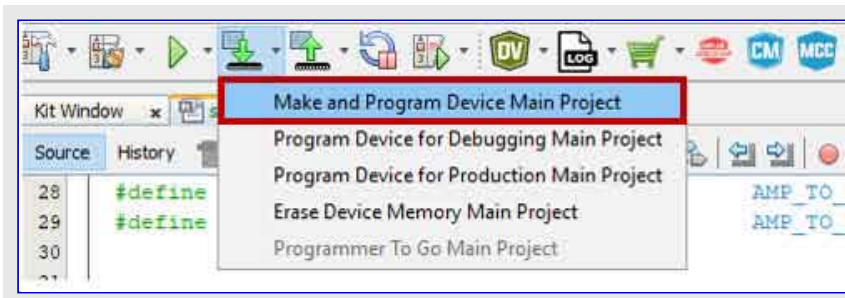
Yes No

10.4. 手順4. 書き込み設定

1. **Clean and Build Project**(解消してプロジェクト構築)をクリックすることによってプロジェクトを構築してください。



2. AVR EB CNANO基板にプロジェクトを書くために**Make and Program Device**(作成してデバイス書き込み)をクリックしてください。

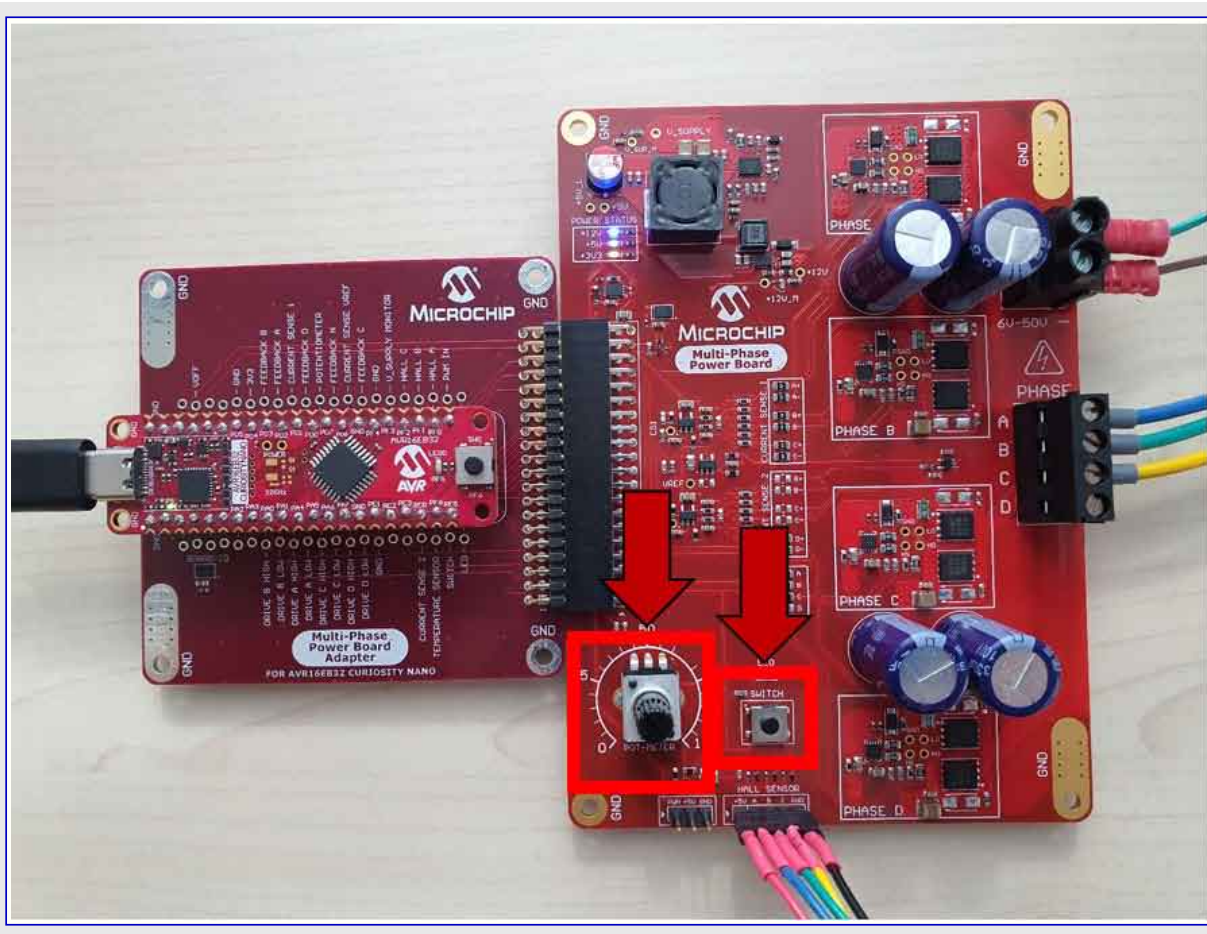


マイクロ コントローラは同期整流(転流)実演が書かれました。

10.5. 手順5. 実演指示

電動機駆動を開始するため、以下の指示に従ってください。

1. 再起動するために釦を長押ししてください。
2. 電動機を開始/停止するには釦を短く押してください。同期動作ではMPPB可変抵抗器を使って振幅が制御されます。



10.6. 手順6. 実演調整

電動機が始動で困難な状況を持つ場合、以下の値の調整を考慮してください。

Motor Startup Voltage [V](電動機始動電圧) – 増加時間の最後で達成された変調された電圧を変更

Ramp-up Time [ms](増加時間) – 変位電圧から始動電圧までを達成するのにかかる時間量

Switch-over Speed [RPM](切替速度) – 増加手順の最後で達する電氣的RPM速度

The screenshot shows a configuration window with two sections: 'Motor Specifications' and 'Start-up Specifications'. In the 'Motor Specifications' section, 'Pole Pairs' is set to 4 and 'Phase Advance [°]' is set to 15. In the 'Start-up Specifications' section, 'Motor Startup Voltage [V]' is set to 3, 'Ramp-up Time [ms]' is set to 1000, 'Ramp-down Time [ms]' is set to 1000, and 'Switch-over Speed [RPM]' is set to 400. The 'Forced Commutation' toggle is turned on.

整流(転流)値を計算するための経験則は次のとおりです。

- VBEMF: 逆起電力(V/kRPM) – 電動機のデータシートから
- SRPM: 定格速度のX% – 電動機のデータシートから
- 増加時間(Ramp-up Time) – 変位電圧から始動電圧までにかかる時間
- 切替速度(Switch-over Speed) – 定格速度のX%

この決まりは良い開始点です。その後、望む結果を得るために調整が必要かもしれません。

開路では電動機の帰還での駆動同期を保証し、電圧振幅(デューティ サイクル)は可変抵抗器によって調整されます。

11. 改訂履歴

文書改訂	日付	注釈
A	2025年2月	初版文書公開

Microchip情報

商標

“Microchip”の名称とロゴ、“M”のロゴ、それと他の名称、ロゴ、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。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。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。

Microchipデバイスコード保護機能

Microchip製品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製品のコード保護機能を侵害する試みは固く禁じられ、デジタルミレニアム著作権法に違反するかもしれません。
- Microchipや他のどの半導体製造業者もそのコードの安全を保証することはできません。コード保護は製品が“破ることができない”ことを当社が保証するということを意味しません。コード保護は常に進化しています。Microchipは当社製品のコード保護機能を継続的に改善することを約束します。

日本語© HERO 2025.

本使用者の手引きはMicrochipのAVR® MCU電動機制御ライブラリ使用者の手引き(DS50003835A-2025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識されている部分もあります。必要に応じて一部加筆されています。頁割の変更により、原本より頁数が少なくなっています。

必要と思われる部分には()内に英語表記や略称などを残す形で表記しています。

青字の部分はリンクとなっています。一般的に赤字の0,1は論理0,1を表します。その他の赤字は重要な部分を表します。