

KMQuick 使い方 編集

もくじ

KMQuick とは

- 搬送システムへの応用

- 直感的なプログラミング

- センサー連動

- タスクセットとは

アップデート履歴

- ver 1.0.6

- ver 1.0.5

必要環境

ダウンロードと実行

基本

- 接続方法

- 接続とタスクの実行

- 遅延処理について

タスクセットの記録と再生

自動化

ファイルの保存と読み込み

センサー入力の使用

- センサーまたはボタンの接続

- インターフェイスの設定

サンプル

- 速度制御

- 位置制御

- トルク制御

- センサー入力による ON/OFF

KMQuick とは

タスクセット（タスク＝KeiganMotor に与える命令のセット）を KeiganMotor 本体に直接記録し、再生するための Windows用ビジュアルプログラミングアプリケーションです。センサーやボタンなどを使用することにより、**他の制御コントローラなしで、簡単な搬送システムなどを構築することができます。**作成したプログラムは、CSV形式で保存と読み込みが可能です。

本ページのPDFは、アプリ本体を含む zip ファイルに含まれます。[ダウンロード](#)を参照下さい。

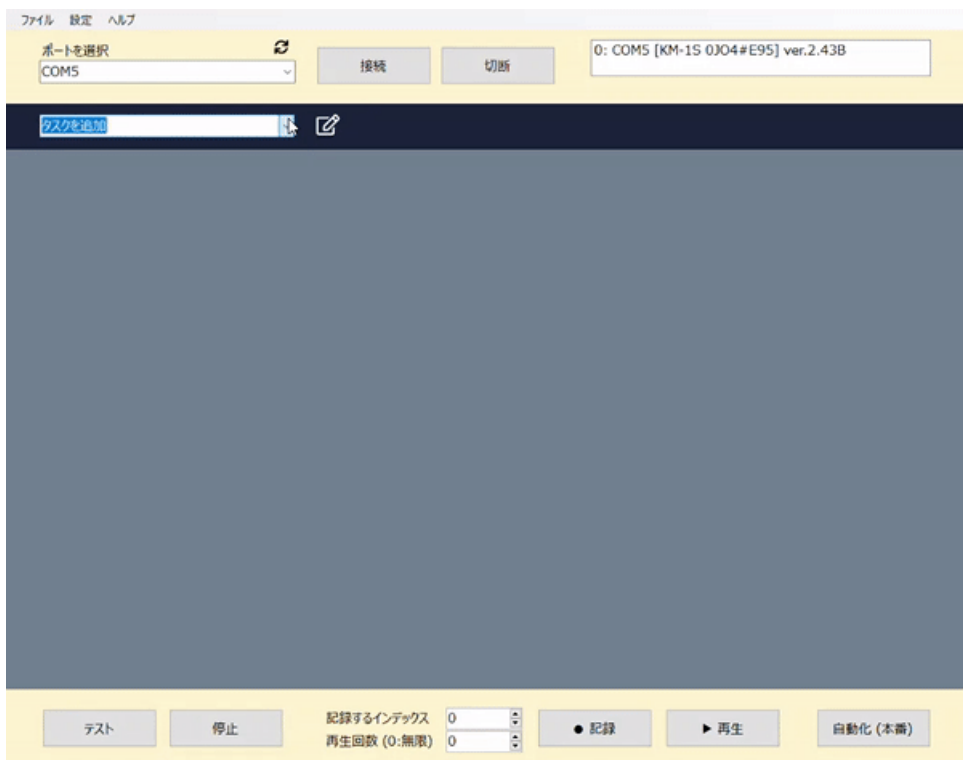
搬送システムへの応用

例えば、以下は KeiganMotor KM-1S-M6829 を使った搬送システムです。KeiganMotor に搬送ローラーを取り付けています。例えば、センサーを検知したときに、回転動作により箱を移動させることができます。このように、[サポート](#)ラなしに、独立搬送システムが構築できます。

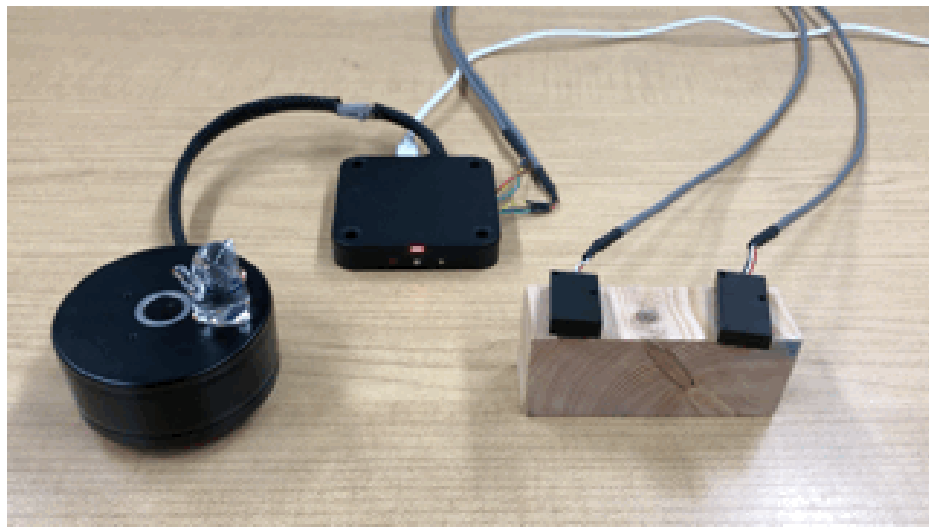


直感的なプログラミング

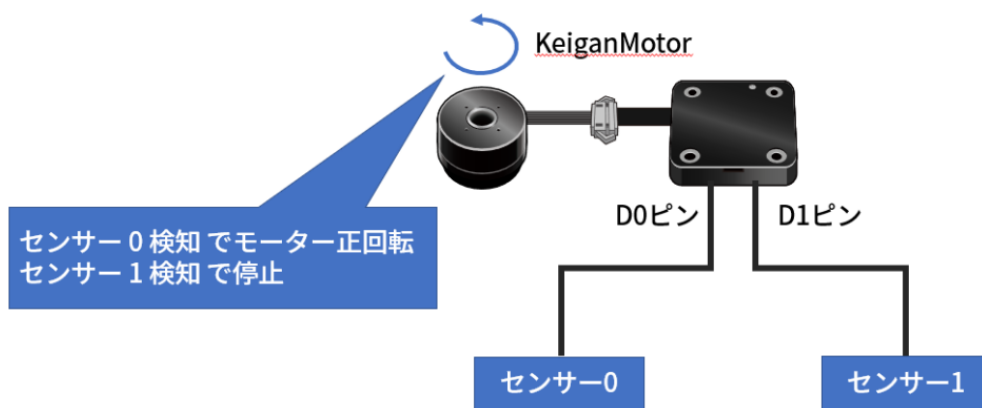
KMQuick は、プログラミングスキルがなくても、メニューから選択するだけで直感的に命令を記録することができます。



センサー連動

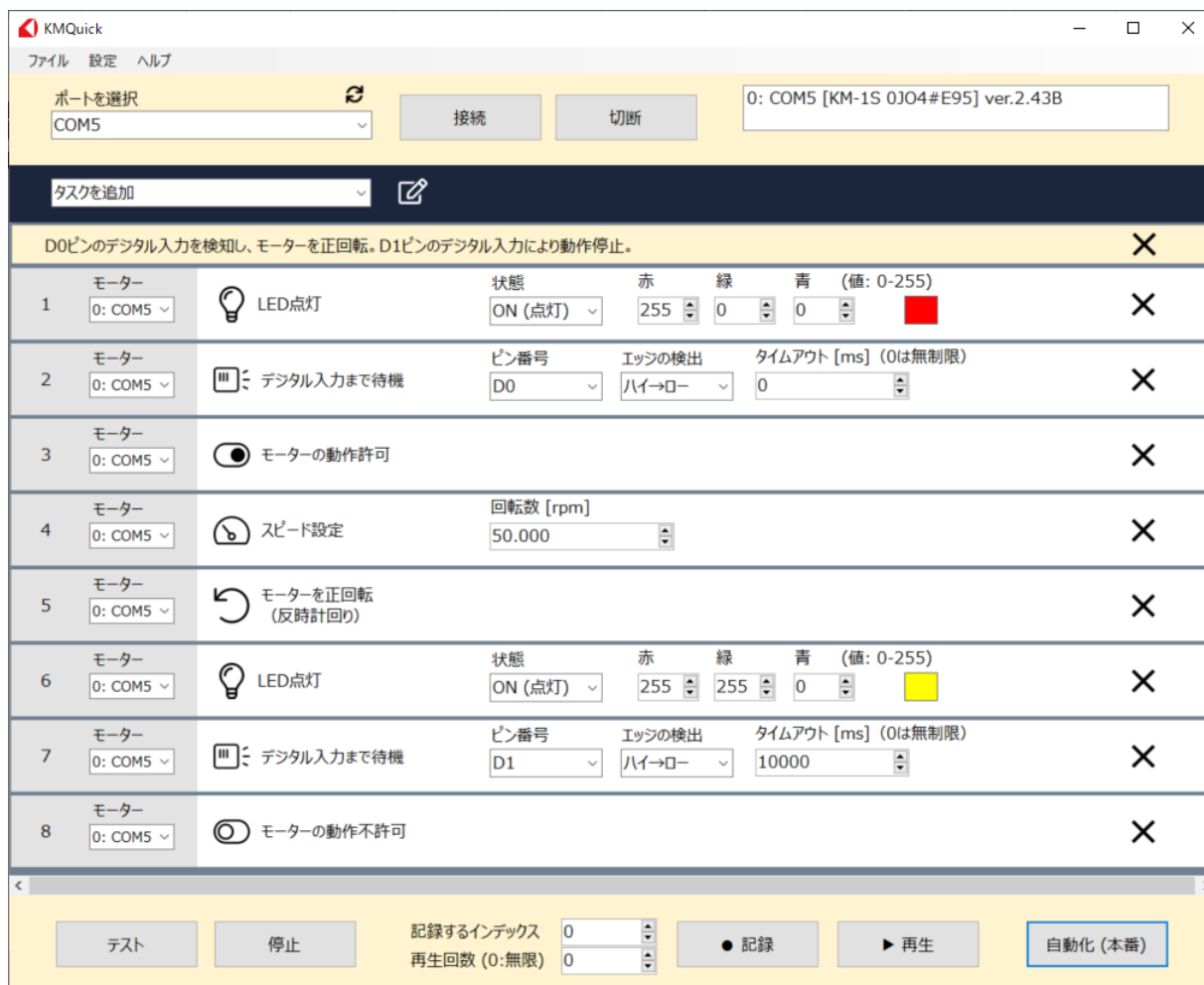


例えば、以下のように構成した場合、2つのセンサーを使用して動作のON/OFFを行うことができます。センサーは、KeiganMotor に直接つながることができます。



タスクセットとは

KeiganMotor では、1つの命令＝コマンドをタスクと称します。KeiganMotor は、複数のタスク＝タスクセットを実行することにより、仕事をこなします。上記の例の場合のタスクセット（複数のタスクのまとまり）は以下のようになります。タスクセットは、KeiganMotor 本体のフラッシュメモリにそのまま記録することができます。



アップデート履歴

ver 1.0.6

詳細は[こちら](#)をご確認下さい。

- ・ タスクセットの消去ボタンを追加しました。
- ・ LEDラジオボタンが反映されない不具合を修正しました。
- ・ LEDラジオボタンのレイアウト崩れ問題を修正しました。

ver 1.0.5

詳細は[こちら](#)をご確認下さい。

- ・ インターフェイスを改善しました。
- ・ 遮光時ON センサーを使用可能にしました。

必要環境

- KeiganMotor KM-1シリーズ ※ ファームウェア ver 2.42 以上
- Windows 10 が動作するPC

※ ファームウェアver. は 2.42 以上が必須です。2.41以下の場合、動作の許可・不許可及び、センサーによる動作ができません。（タスクセットへの保存を可能とする、新設のコマンドを使用しているため）こちらから 2.43（ベータ版）をダウンロードしてアップデートして下さい。

[km1_firmware_ver_243B](#) [ダウンロード](#)

[ファームウェアのアップデート方法はこちらを参照下さい。](#) スマートフォンから指定のアプリを経由してアップデートします。

ダウンロードと実行

以下から KMQuick 実行ファイルを以下からダウンロードして下さい。ダウンロードして解凍したフォルダを任意の場所に保存します。

※ .exe 以外のファイルも削除しないで下さい。解凍したフォルダ/examples フォルダ内にサンプルのCSVファイルがあります。

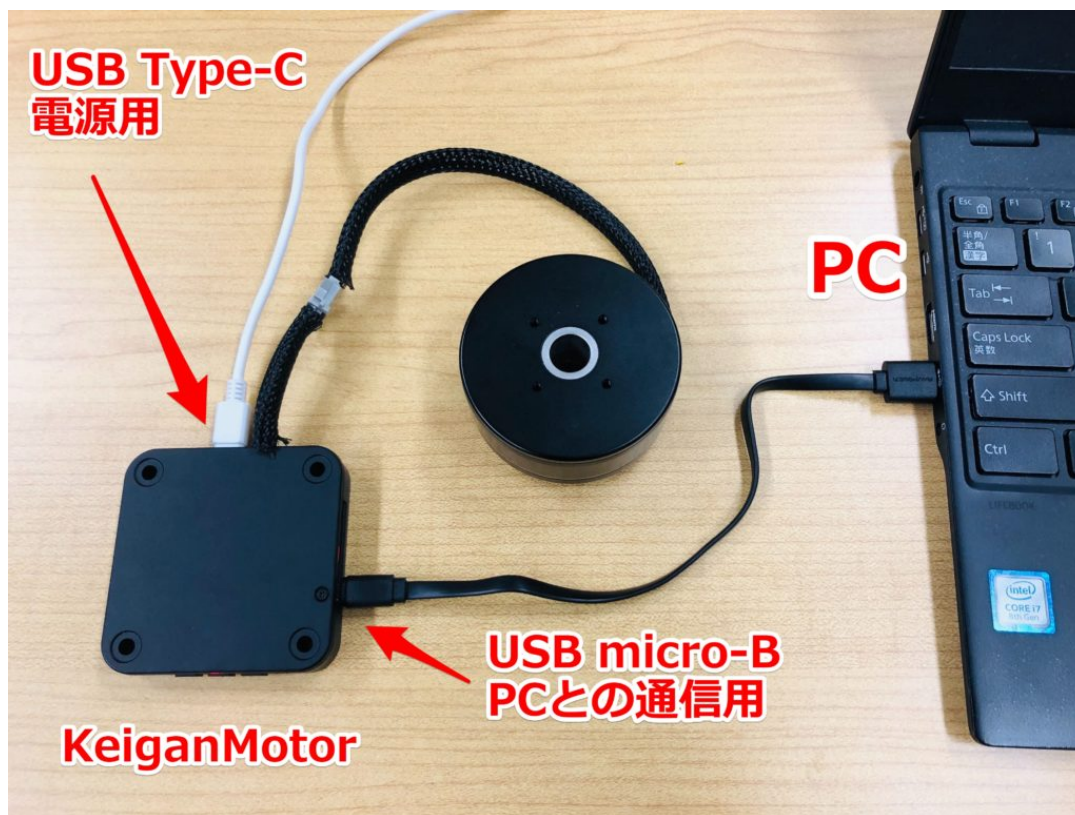
[KMQuick_1.0.5](#) [ダウンロード](#)

名前		更新日時	種類	サイズ
examples	サンプル集	2020/08/25 11:27	ファイル フォルダー	
FontAwesome.Sharp.dll		2020/06/07 15:48	アプリケーション拡張	439 KB
KeiganMotor_USB.dll		2020/08/17 17:40	アプリケーション拡張	35 KB
KeiganMotor_USB.pdb		2020/08/17 17:40	Program Debug Database	104 KB
KMQuick.exe	ダブルクリック	2020/08/25 11:24	アプリケーション	388 KB
KMQuick.exe.config		2020/08/01 13:35	Configuration ソース ファ...	1 KB
KMQuick.pdb		2020/08/25 11:24	Program Debug Database	262 KB
KMQuick_使い方.pdf	使い方（本ページの内容PDF）	18:25	PDF-XChange Viewer D...	3,466 KB

保存後、フォルダ内の `KMQuick.exe` をダブルクリックして実行してください。（examples フォルダには、[サンプル集](#) が格納されています）

基本

接続方法



USB ケーブルを使用して、上図のように、PC と KeiganMotor を接続して下さい。KeiganMotor には、電源用と、PC との通信用の2種類のUSBケーブルを接続して下さい。特に、通信用の USB Type-A to micro B ケーブルは、KeiganMotor 本体に同梱されていないので、ご準備下さい。

KeiganMotor のUSBポートへのアクセス方法は、以下を参照下さい。

[有線ポートについて](#)

接続できない場合

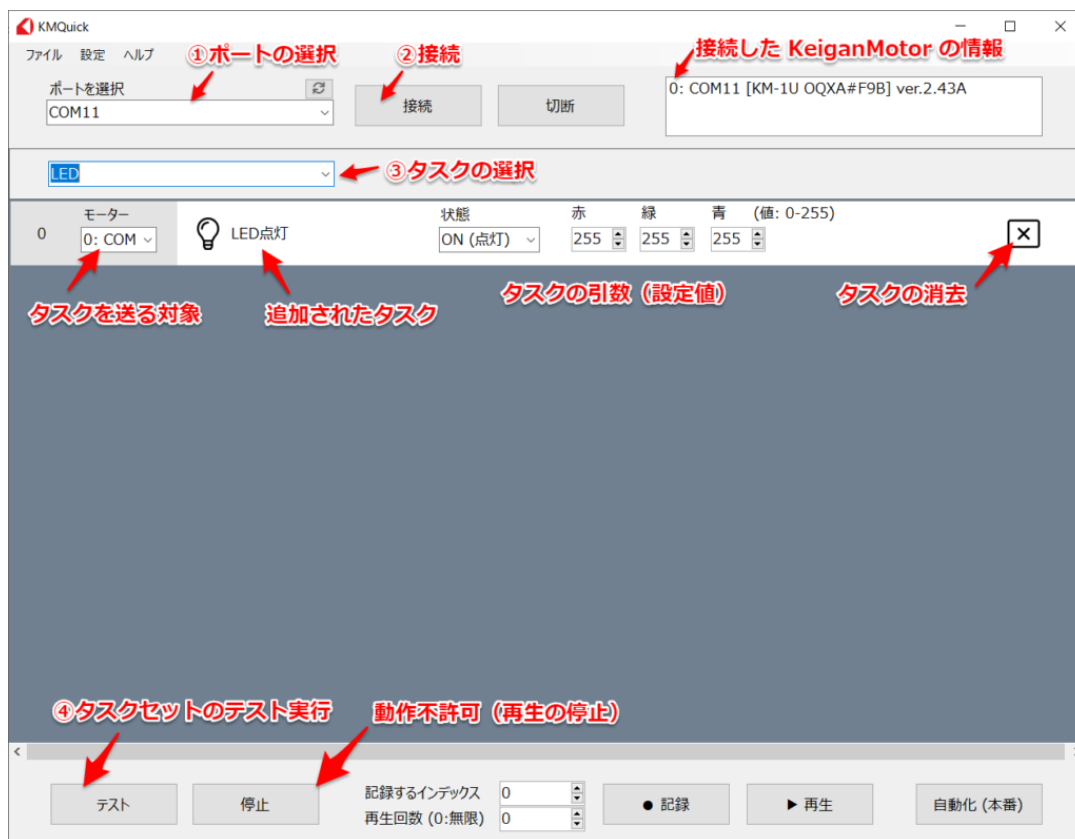
PCとモーターを接続してもプルダウンにCOMポートの選択肢が表示されない場合、以下をご確認下さい。

1. [Windows の仮想COMドライバを再インストール](#)する
2. 通信用の USB Type-A to micro B ケーブルを交換する
3. [本体を工場出荷状態にリセット](#)する

上記で解決しない場合、コネクタが破損している可能性があります。「サポート」よりお問い合わせ下さい。

接続とタスクの実行

以下の手順でLEDを点灯させるタスクセットを実行してみましょう。



(1) ポートの選択と接続

ポート番号のリストから、KeiganMotor を接続した COMポートを選択し、接続ボタンを押すことによりUSBデバイスとして接続します。正常に接続された場合、右上のリストボックスに、以下の情報が表示されます。

接続番号: COMポート名 [デバイス名] ファームウェアバージョン

(2) タスクの選択

KeiganMotor に送るコマンド = タスクを選択します。今回は、KeiganMotor 本体のLEDを点灯させるタスク [LED] を選択します。

タスクをプルダウンメニューから選択すると、タスクセットエディタ（画面上のグレーの領域）にタスクが追加されます。

追加したタスクには、以下の情報が必要です。

- タスクを送る対象のKeiganMotor（接続番号とCOMポート名）
- 引数（タスクに対応する設定値＝スピードやLEDの輝度などの数値）

KeiganMotorが接続されている場合、対象の KeiganMotor は自動的に選択されます。タスク [LED] の引数は、デフォルトでは、以下のようになります。

(3) タスクセットのテスト実行

ON（点灯状態）, 各カラーの輝度 R: 255, G: 255, B: 255

タスクを設定した状態で、左下の [テスト] ボタンをクリックすると、タスクセットとして登録した一連のタスクが再生されます。KeiganMotor 本体のLEDが変わることを確認してください。今度は、試しにカラーを変更してみてください。

基本的な操作方法是以上です。本アプリには、CSVファイルで作成されたサンプルが含まれます。速度制御や位置制御をすぐに試したい方は、以下を参考に実行してみてください。

[ファイルの保存と読み込み](#)

[サンプル](#)

遅延処理について

次は、以下のようなタスクセットを作成して見ましょう。

モーターが 5秒間回転し、停止します。

KMQuick

ファイル 設定 ヘルプ

ポートを選択COM11

接続

切断

0: COM11 [KM-1U OQXA#F9B] ver.2.43A

停止

0	モーター 0: COM	LED点灯	状態 ON (点灯)	赤 255	緑 255	青 (値: 0-255) 255	×
1	モーター 0: COM	モーターの動作許可					×
2	モーター 0: COM	スピード設定	回転数 [rpm] 10.000				×
3	モーター 0: COM	モーターを正回転 (反時計回り)					×
4	モーター 0: COM	次のタスク実行まで 指定時間待機	待機時間 [ms] 5000				×
5	モーター 0: COM	モーターの停止	状態 速度ゼロ (トルクあり)				×

テスト

停止

記録するインデックス
再生回数 (0:無限)

0
0

● 記録

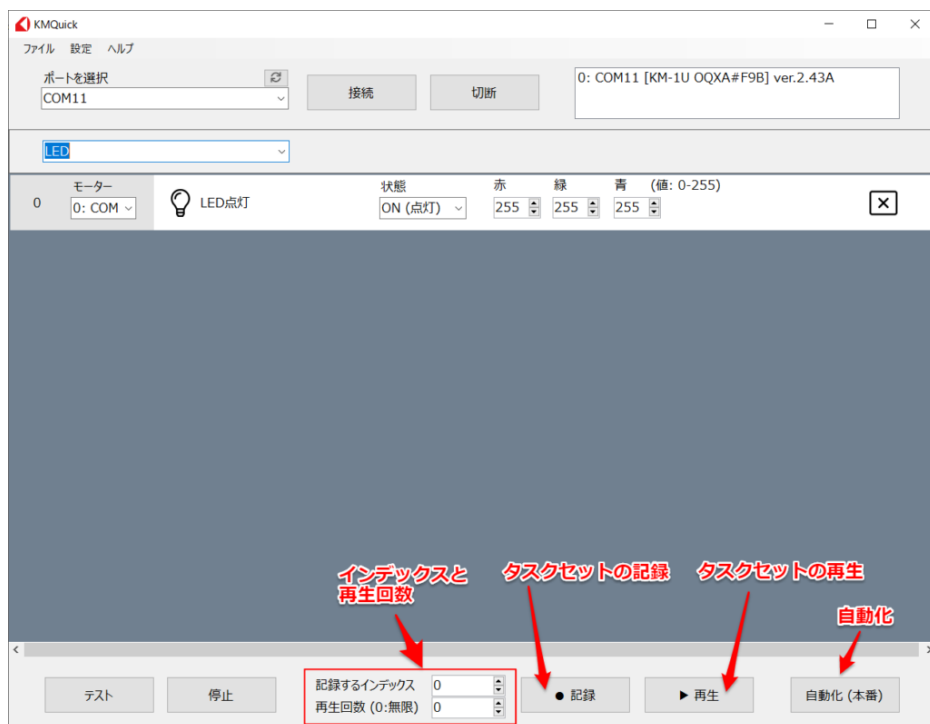
▶ 再生

自動化 (本番)

[次のタスク実行まで指定時間待機] を使用することにより、次のタスク実行を、一定時間待機させることができます。本タスクにより、KeiganMotor の内部キュー（先入れ先出しのコマンド処理構造）一定時間停止し、次のコマンドの実行を遅延させます。

タスクセットの記録と再生

登録したタスクセットを、KeiganMotor 本体のフラッシュメモリに記録することができます。



KM-1 シリーズでは、0から49までのインデックスで区別される、合計50個のタスクセットを記録可能です。画面下から [記録するインデックス] と [再生回数] を設定して下さい。再生回数に 0 をセットすると、無限回再生となります。

[記録] ボタンをクリックすると、タスクセットが本体に保存されます。フラッシュアクセス中は、本体のLEDが緑白に点滅します。

[再生] ボタンをクリックすると、本体に保存されたタスクセットが再生されます。タスクセットが存在しない場合、KeiganMotor 本体のLEDが一時的に赤青に点滅します。

右下の [自動化] ボタンをクリックすることにより、KeiganMotor を起動した際に、記録したタスクセットを自動再生するための設定が可能となります。

自動化

KeiganMotor の電源投入後、記録されたタスクセットを自動で実行するための設定を行います。また、ボタン押下で実行をスタートすることもできます。これにより、外部の制御コントローラを必要とせず、反復動作をするシステムを構築することができます。再生回数は、前項で設定した回数を継承します。

[自動化] ボタンをクリックすると、下記のウィンドウが立ち上がります。

無線 (Bluetooth Low Energy) は、環境に応じて [設定] ボタンにより無効化して下さい。特に、第三者が近くで無線を使用する恐れのある場合は、無効化することを推奨します。



記録したタスクセットの自動化には、以下のオプションがあります。

- 再起動後、すぐに開始
- 再起動後、本体 ▶（再生）ボタンで開始
- 自動化を停止

いずれかをトグルボタン（左側の黒丸で選択するボタン）で選択後、[実行] ボタンを押してください。KeiganMotor が自動で再起動し、設定した自動化が開始されます。[再起動後、すぐに開始] 設定の場合、**タスクセットの内容により、自動で回転動作を開始しますので、安全にご注意下さい。**

ファイルの保存と読み込み

作成したタスクセットは、CSVファイルとして保存できます。

[ファイル] メニューをクリックすると、[保存] と [読み込み] が可能です。[保存] メニューをクリックした場合、デスクトップなどの任意のフォルダに、csv ファイル名を指定して保存することができます。



保存したCSVファイルは、[読み込み] メニューからロードすることができます。また、**CSVファイルを開いて、直接編集することもできます。** 引数が間違っていた場合、読み込みエラーとなる場合がありますので、ご注意ください。

	A	B	C	D	E	F	G	H
1	No.	モーター	タスク名	引数1	引数2	引数3	引数4	引数5
2	0	0	LED	1	255	0	0	
3	1	0	動作許可					
4	2	0	スピード設定	50				
5	3	0	正回転					
6	4	0	指定時間待機	3000				
7	5	0	LED	1	0	255	0	
8	6	0	逆回転					
9	7	0	指定時間待機	3000				
10	8	0	動作不許可					

保存されたCSVファイルの例

センサー入力の使用

デジタルセンサー入力を KMQuick で使用する場合、KeiganMotor本体のファームウェア ver. 2.42 以上が必要となります。以下の手順により、センサーの接続を行ってください。

センサーまたはボタンの接続

[こちら](#)を参照して、I2Cポートにセンサーやボタンを接続して下さい。センサーの条件は、以下を満たすものとなります。

- 5Vで動作すること
- NPN出力型であること。（入力されるデジタル信号 ハイ（3.0V）または ロー（0V）を検知する）

D0, D1 の信号入力は、5V トレラントではありません。5Vの信号入力は KeiganMotor 電子基板の故障に繋がります。必ず 3.3V 以下の入力電圧になるようにして下さい。NPN出力型のセンサーをお使いいただければ、問題ありません。

物体検知に最適な反射型の光電センサーとして、センサテック社の以下のものを推奨します。ver 1.0.5 以降、遮光時ONのセンサーにも対応可能になりました。

センサテック社 拡散反射形光電センサ

https://www.sensatec.co.jp/products/detail.php?product_id=220

形式	ver 1.0.1	ver 1.0.5
PRA-300-5（入光時ON / 30mm）	○	○
PRA-500-5（入光時ON / 50mm）	○	○
PRA-101-5（入光時ON / 100mm）	○	○
PRA-201-5（入光時ON / 200mm）	○	○
PRA-300-51（遮光時ON / 30mm）	非対応	○
PRA-500-51（遮光時ON / 50mm）	非対応	○
PRA-101-51（遮光時ON / 100mm）	非対応	○

形式	ver 1.0.1	ver 1.0.5
PRA-201-51（遮光時ONN / 200mm）	非対応	○

※ 他のセンサーもお使い頂けますが、NPN型センサー（オープンドレイン）かつ、5Vで動作する必要があります。

本センサーと、KeiganMotor のデジタル入力 D0, D1 に簡単に接続するためのケーブルを提供可能です。詳細はお問い合わせ下さい。

ワークを検知した際に（遮光された状態で）動作させる場合、以下のようになります。

入光時ON（遮光時OFF）センサーの場合

入力信号（エッジの検出）を「ハイ→ロー」として下さい。

遮光時ON（入光時OFF）センサーの場合

入力信号（エッジの検出）を「ロー→ハイ」として下さい。

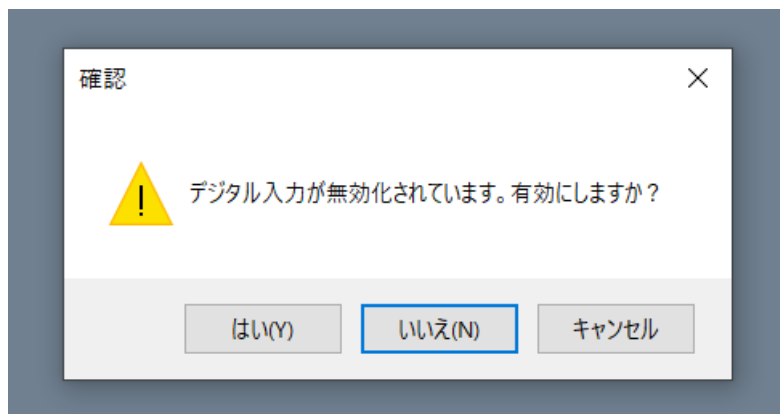
センサーを利用したサンプルコードは、examplesフォルダ内の sensor.csv を参照下さい。※

インターフェイスの設定

I2Cポートをデジタル入力として使用する場合には、デフォルト状態からインターフェイス設定を以下のように変更する必要があります。

I2C通信: 無効、デジタル入力: 有効

登録したタスクセットに、[デジタル入力まで待機] がある場合で、上記設定になっていない場合、テスト実行時やタスクセット再生時に、以下のダイアログが自動で表示されます。[はい] を選択することにより、上記の設定がフラッシュに保存され、デジタル入力を使用可能となります。



上記より、アプリケーションをダウンロードした際のフォルダ内に、examples というフォルダがあります。examples フォルダ内には、KMQuick で読み込み可能なCSVファイルのサンプル例がありますので、すぐにお使い頂けます。CSVファイルの読み込み方法は[こちら](#)を参照ください。

速度制御

run.csv

スピード 50rpm で3秒間正回転し、6秒間逆回転します。

位置制御

絶対位置

moveto.csv

スピード 50rpm で絶対位置 180° に移動後、絶対位置 -180°に移動します。（移動+位置決め時間はそれぞれ3秒の設定です）

ヒント：[現在位置を設定] タスクを前に挿入することにより、原点設定が可能です。

相対位置

moveby.csv

スピード 50rpm で相対位置 90° に移動後、相対位置 -90°（元の場所）に移動します。（移動+位置決め時間はそれぞれ3秒の設定です）

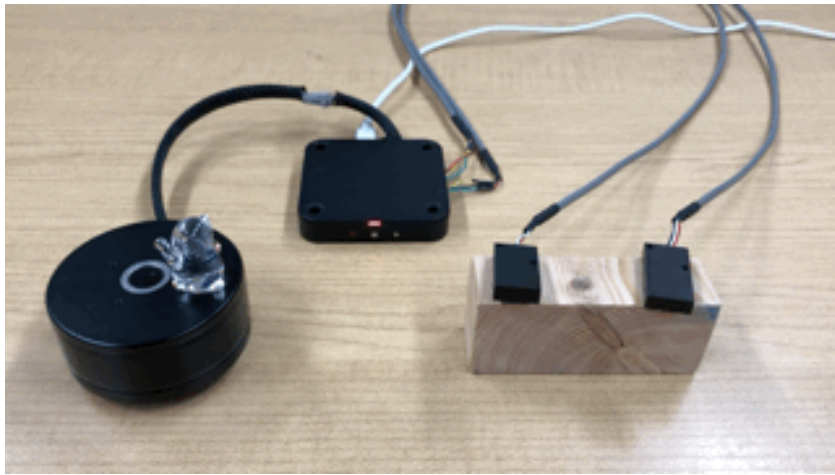
トルク制御

torque.csv

スピード 50rpm 最大トルク 0.1 Nm で正回転します。安全装置や、押し当て制御等に応用できます。

センサー入力による ON/OFF

sensor.csv



遮光時OFFとなるセンサーを用いて、デジタル入力 D0 が ハイ→ローに変化した場合、正回転を開始します。正回転中、デジタル入力 D1 が ハイ→ローに変化あるいは10秒経過した場合、停止します（動作不許可）。

※ 遮光時ONとなるセンサーの場合は、ロー→ハイ 検知に書き換えて下さい。

[◀【前の記事】KMQuick](#)

[KM Quick ダウンロード](#) [【次の記事】▶](#)

まずはじめに

▣ はじめにお読み下さい

▣ 準備と基本

▣ 高度な使い方

お問合せ

サポートセンター

アプリ

▣ KMQuick

[KMQuick 使い方](#)

[KM Quick ダウンロード](#)

▣ KeiganPlay

▣ KeiganCore

[sample] Webアプリ

ファームウェア

ファームウェアのアップデート方法

ファームウェアのダウンロード

ソフトウェア

ソフトウェア開発の概要

有線ポートについて

▣ BLE通信

▣ ローレベルAPI

<div><div></div><div>Arduino / ESP</div></div>
<div><div></div><div>Python</div></div>
<div><div></div><div>Node.js Javascript</div></div>
<div><div></div><div>(旧)Node-RED</div></div>
<div><div></div><div>Windows C#</div></div>
<div><div></div><div>Swift</div></div>
<div><div></div><div>ROS</div></div>
<div><div></div><div>KeiganPi</div></div>
<div><div></div><div>KeiganPi</div></div>
<div><div></div><div>キット・オプション</div></div>
<div><div></div><div>KM-1 キット・オプション</div></div>
<div><div></div><div>KeiganPi</div></div>