

概要 更新履歴	KM2IO20Aki_IOBit.hsp		
	USB-IO2.0(AKI) IOBitドライバー		
	更新日時	20240219	(c)2023-2024 team.Drakuji

●注意

このプログラムおよび動作環境においていかなる事故・損害に対する補償はできません。

●概要

Km2Net株式会社様のUSB-IO2.0(AKI)I/Oモジュールを、IOBit定義仕様に基づいて制御する命令関数群です。

◆最大接続数	16モジュール		
◆対応環境	Windows版HSP		
◆必須ファイル	#include	usb-ioFamily.as	
	#uselib	user32.dll	

モジュールナンバーとモジュールアドレスについて

モジュールナンバーは、プラットフォームOSが認識した順番に割り振られるモジュールごとに識別番号です。
モジュールアドレスは、J2_0からJ2_3ポートの入力状態をモジュールアドレス番号として識別します。

IOBit名前関数について

他のI/Oモジュールと共通する命令や関数には、IOBitの名前を使用した共通の名前が使用できます。

プルアップについて

J2ピン群のみプルアップ抵抗の有効無効が指定できます。

●更新履歴

日時	内容
▼20240219	初版
	初版リリース

設定 定数定義	KM2IO20Aki_IOBit.hsp		
	USB-IO2.0(AKI) IOBitドライバー		
	更新日時	20240219	(c)2023-2024 team.Drakuji

●設定

※先頭に"_"のあるものは、ローカル定義・定数です。

設定名	定数	説明
<hr/>		
<u>_USEMOD</u>		
モジュールを使用するか設定します。		
	0 使用しない	
	1 使用する	
<hr/>		
<u>_USEADR</u>		
J2ポートをモジュールアドレス識別として使用するか設定します。		
	0 使用しない	
	1 使用する	
<hr/>		
<u>_DIALOG</u>		
警告ダイアログを表示するか設定します。		
	0 表示しない	
	1 表示する	
<hr/>		
<u>_USEIOBIT</u>		
IOBit名前関数を使用するか設定します。		
	0 使用しない	
	1 使用する	

●定数定義

※先頭に"_"のあるものは、ローカル定義・定数です。

定数定義名	定数・定義	説明
<hr/>		
<u>KM2IO20AKI_NAME</u>		
ドライバーの名称です。		
	“KM2IO20AKi”	
<hr/>		
<u>KM2IO20AKI_VER</u>		
ドライバーのバージョンです。		
	“バージョン番号”	
<hr/>		
<u>KM2IO20AKI_STAT_?</u>		
タスク定義です。		
OK	0 正常	
NG	1 不明エラー	
NG_CON	2 接続エラー	
NG_ADR	3 モジュールアドレス指定エラー	
NG_DB	4 モジュールアドレス重複エラー	
NG_SYNC	5 通信エラー	
NG_CFG	6 設定エラー	
NG_MODE	7 ピンモードエラー	
<hr/>		
<u>KM2IO20AKI_PINS / IOBIT_PINS</u>		
モジュール1台あたりの入出力ピン数です。		
	12 ピン数	
<hr/>		
<u>KM2IO20AKI_PIN_?</u>		
モジュール表記のピン定義名です。		
J1_0	0 ピン名J1_0	
J1_1	1 ピン名J1_1	
J1_2	2 ピン名J1_2	
J1_3	3 ピン名J1_3	
J1_4	4 ピン名J1_4	

J1_5	5 ピン名J1_5
J1_6	6 ピン名J1_6
J1_7	7 ピン名J1_7
J2_0	8 ピン名J2_0
J2_1	9 ピン名J2_1
J2_2	10 ピン名J2_2
J2_3	11 ピン名J2_3

KM2IO20AKI_PULLUP_?

プルアップ設定の定義です。

ENABLE	0 有効
DISABLE	1 無効

機能 命令関数群	KM2IO20Aki_IOBit.hsp		
	USB-IO2.0(AKI) IOBitドライバー		
	更新日時	20240219	(c)2023-2024 team.Drakuji

●コマンド命令・関数一覧

命令・関数	引数名・戻り値	型	説明
InitKM2IO20Aki modules / InitIOBit modules			
I/Oモジュールを初期化接続します。必ず最初に実行してください。			
modules	int	接続モジュール数(0から16)	※モジュールアドレスを使わない時、0を指定すると接続数を自動で判別 タスク値(KM2IO20AKI_STAT_?)
stat	int		
HaltKM2IO20Aki / HaltIOBit			
I/Oモジュールを接続解除します。必ず最後に実行してください。			
※出力は全てLoになります。			
stat	int	タスク値(KM2IO20AKI_STAT_?)	
ReadKM2IO20AkiCfg			
接続されている全てのI/Oモジュールの設定データを読み込みます。			
stat	int	タスク値(KM2IO20AKI_STAT_?)	
WriteKM2IO20AkiCfg modnum			
I/Oモジュールに設定データを書き込みます。			
modnum	int	モジュールナンバー	タスク値(KM2IO20AKI_STAT_?)
stat	int		
WriteKM2IO20AkiCfgByAdr modadr			
I/Oモジュールに設定データを書き込みます。			
modadr	int	モジュールアドレス	タスク値(KM2IO20AKI_STAT_?)
stat	int		
GetKM2IO20AkiCnt() / GetIOBitCnt()			
I/Oモジュールの接続数を返します。			
return	int	接続台数	
GetKM2IO20AkiUseAdr			
モジュールアドレス使用設定の状態を返します			
return	int	無効:0 有効:1	
GetKM2IO20AkiModNum(modadr)			
モジュールアドレスのモジュールナンバーを取得します。			
modadr	int	モジュールアドレス	※無効な場合-1が返ります。
return	int	モジュールナンバー	
GetKM2IO20AkiModAdr(modnum)			
モジュールアドレスのモジュールナンバーを取得します。			
modnum	int	モジュールナンバー	※無効な場合-1が返ります。
return	int	モジュールアドレス	
SetKM2IO20AkiPullUp modnum,pullup			
J2-0からJ2-3ピンのプルアップを設定します。			
modnum	int	モジュールナンバー	タスク値(KM2IO20AKI_STAT_?)
pullup	int	モジュールアドレス	
stat	int		
SetKM2IO20AkiPullUpByAdr modadr,pullup			
J2-0からJ2-3ピンのプルアップを設定します。			
modadr	int	モジュールアドレス	モジュールアドレス
pullup	int		

stat	int	タスク値(KM2IO20AKI_STAT_?)
------	-----	-------------------------

GetKM2IO20AkiPullUp(modnum)		
J2-0からJ2-3ピンのプルアップ設定を取得します。		
modnum	int	モジュールナンバー
return	int	有効:0 無効:1

GetKM2IO20AkiPullUpByAdr(modadr)		
J2-0からJ2-3ピンのプルアップ設定を取得します。		
modadr	int	モジュールアドレス
return	int	有効:0 無効:1

GetKM2IO20AkiStatus() / GetIOBitStatus()		
I/Oモジュール状態を文字列で返します。		
return	str	状態文字列

GetKM2IO20AkiPinStr(pin) / GetIOBitPinStr(pin)		
I/Oモジュールのピン名を返します。		
pin	int	ピン通し番号
return	str	ピン名

KM2IO20AkiMain / IOBitMain		
I/Oモジュールメイン関数です。状態を更新する度に実行してください。		
stat	int	タスク値(KM2IO20AKI_STAT_?)

SetPinMode modnum,pin,mode		
ピンモードを設定します。		
modnum	int	モジュールナンバー
pin	int	ピン定義名
mode	int	ピンモード定義名(IOBitを参照) ※D_OUT、D_IN、A_OUTのみ
stat	int	タスク値(KM2IO20AKI_STAT_?)

SetPinModeByAdr modadr,pin,mode		
ピンモードを設定します。		
modadr	int	モジュールアドレス
pin	int	ピン定義名
mode	int	ピンモード定義名(IOBitを参照) ※D_OUT、D_IN、A_OUTのみ
stat	int	タスク値(KM2IO20AKI_STAT_?)

GetPinMode(modnum,pin)		
現在のピンモードを取得します。		
modnum	int	モジュールナンバー
pin	int	ピン定義名
return	int	ピンモード定義名(IOBitを参照)

GetPinModeByAdr(modadr,pin)		
現在のピンモードを取得します。		
modadr	int	モジュールアドレス
pin	int	ピン定義名
return	int	ピンモード定義名(IOBitを参照) ※異常な場合PINMODE_OTHERが返ります。

SetPinPWM modnum,pin,freq,duty		
A_OUTピンに疑似PWM制御出力を設定します。		
modnum	int	モジュールナンバー
pin	int	ピン定義名
freq	int	周期(0から255)
duty	int	周期のうちHiレベルにする時間
stat	int	タスク値(KM2IO20AKI_STAT_?)

SetPinPWMBByAdr modadr,pin,freq,duty

A_OUTピンに疑似PWM制御出力を設定します。

modadr	int	モジュールアドレス
pin	int	ピン定義名
freq	int	周期(0から255)
duty	int	周期のうちHiレベルにする時間
stat	int	タスク値(KM2IO20AKI_STAT_?)

SetPinState modnum,pin,lv

D_OUTピンの出力レベルを設定します。

modnum	int	モジュールナンバー
pin	int	ピン定義名
lv	int	ピン状態(レベル)定義名または(Hi:1 Lo:0)
stat	int	タスク値(KM2IO20AKI_STAT_?)

SetPinStateByAdr modadr,pin,lv

D_OUTピンの出力レベルを設定します。

modadr	int	モジュールアドレス
pin	int	ピン定義名
lv	int	ピン状態(レベル)定義名または(Hi:1 Lo:0)
stat	int	タスク値(KM2IO20AKI_STAT_?)

GetPinState(modnum,pin)

ピンの状態(レベル)を取得します。

modnum	int	モジュールナンバー
pin	int	ピン定義名
return	int	ピン状態(レベル)(Hi:1 Lo:0)

GetPinStateByAdr(modAdr,pin)

ピンの状態(レベル)を取得します。

modadr	int	モジュールアドレス
pin	int	ピン定義名
return	int	ピン状態(レベル)(Hi:1 Lo:0)

GetPinData(modnum,pin,pindata)

ピンデータを取得します。

modnum	int	モジュールナンバー
pin	int	ピン定義名
pindata	int	ピンデータ要素定義名
return	int	ピンデータ要素の回答

GetPinDataByAdr(modadr,pin,pindata)

ピンデータを取得します。

modadr	int	モジュールアドレス
pin	int	ピン定義名
pindata	int	ピンデータ要素定義名
return	int	ピンデータ要素の回答

※異常な場合-1が返ります。

付録	KM2IO20Aki_IOBit.hsp		
	USB-IO2.0(AKI) IOBitドライバー		
	更新日時	20240219	(c)2023-2024 team.Drakuji

●ピンの定義

ピン番号	ピン名	機能	特記事項
0	J1_0	D_I/O	
1	J1_1	D_I/O	
2	J1_2	D_I/O	
3	J1_3	D_I/O	
4	J1_4	D_I/O	
5	J1_5	D_I/O	
6	J1_6	D_I/O	
7	J1_7	D_I/O	
8	J2_0	D_I/O	ブルアップの有効無効,モジュールアドレス指定
9	J2_1	D_I/O	ブルアップの有効無効,モジュールアドレス指定
10	J2_2	D_I/O	ブルアップの有効無効,モジュールアドレス指定
11	J2_3	D_I/O	ブルアップの有効無効,モジュールアドレス指定

●モジュールナンバー(数え番号)とモジュールアドレス

CPUに最大16台のI/Oモジュールを接続することができ、制御することができます、
I/Oモジュールの仕様上、各I/Oモジュールを認識する順番がUSBハブなどにより使用する環境によって不定になります。
この問題を解決するためにJ2ピン群を使用して初期化後にモジュールアドレスを割り振ることで
確実に指定したI/Oモジュールに対して入出力の制御ができます。

▼複数のI/Oモジュールを区別する数字の違い

- ◆モジュールナンバー・・・
- 初期化時にI/Oモジュールを認識した順番に0から割り振られる順不同な番号
- ◆モジュールアドレス・・・
- 初期化時にI/OモジュールのJ2ピン群のジャンパ状態をビット表現した値

▼モジュールアドレス

接続数	アドレス予約ピン	アドレス	ジャンパするピン
1	J2_0	0	なし
2		1	J2_0
3	J2_0 J2_1	2	J2_1
4		3	J2_1,J2_0
5	J2_0 J2_1 J2_2	4	J2-2
6		5	J2_2,J2_0
7		6	J2_2,J2_1
8		7	J2_2,J2_1,J2_0
9	J2_0 J2_1 J2_2 J2_3	8	J2_3
10		9	J2_3,J2_0
11		10	J2_3,J2_1
12		11	J2_3,J2_1,J2_0
13		12	J2_3,J2_2
14		13	J2_3,J2_2,J2_0
15		14	J2_3,J2_2,J2_1
16		15	J2_3,J2_2,J2_1,J2_0

- ◆接続数
- 接続認識したI/Oモジュールの数
- ◆アドレス予約ピン
- 接続数に応じてアドレス番号を決めるためのピン名
- ※このピンは予めピンモードを入力(D_IN)に設定しておく必要があります。
- ※接続数が4台の場合は、J2_0,J2_1の2つのピンを使用して判定します。
- 残りのJ2_2,J2_3は自由に使用できるピンです。
- ※入力ブルアップの有効無効はどちらにも対応します。
- ◆アドレス
- アドレス番号
- ◆ジャンパするピン
- アドレス番号を指定するためにジャンパするピン名
- ※必ずジャンパしたモジュールアドレス値が正しく反映されているか確認してください。

▼モジュールアドレスが"-1"になる場合の確認

- ◆モジュールアドレスが重複している、または同じ入力ジャンパ状態のI/Oモジュールがある
- 割り当て処理を取り消しされ、全モジュールアドレスは"-1"になります。

- ◆アドレス予約ピンモードが入力(D_IN)でないとき
該当モジュールアドレスは割り振られず、"-1"になります。

▼命令や関数をモジュールアドレスで指定する

- ◆GetKM2IO20AkiModNum例
GetKM2IO20AkiModNum(モジュールアドレス)
※↑でモジュールナンバーを確認取得して控えておき、関数で使います。
- ◆接尾に"ByAdr"のある命令や関数の例
GetPinStateByAdr(モジュールアドレス,ピン番号)
※↑ByAdrのある命令や関数は、I/Oモジュールの識別をモジュールアドレスで指定します。
※不正なI/Oモジュールを指定した場合は命令や関数は実行されず、
stat戻値のあるものは初期値またはKM2IO20AKI_STAT_NGが返されます。

●ピンモードの確認と変更

指定ピン番号のピンモードがこれから使うピンモードと異なるとき、ピンモードを変更する必要があります。
ピンモードの変更が不要と分かっている場合でも、これから使うピンモードを照合しておくことをおすすめします。

▼ピンモードの確認方法

- ◆GetPinMode例
if GetPinMode(モジュールナンバー,ピン番号) != 使う用途のピンモード番号{
 dialog "ピンモードが異なります!"
}
- ◆SetPinMode例
SetPinMode モジュールナンバー,ピン番号,使う用途のピンモード番号
if stat == 0 : dialog "ピンモードが異なります!"

▼ピンモードの変更方法

- ◆SetPinMode
↓
↓
SetPinMode モジュールナンバー,ピン番号,使う用途のピンモード番号
↑ この処理を変更が必要なピン全てで行います。
- ◆SetKM2IO20AkiPullUp
↓
↓
SetKM2IO20AkiPullUp モジュールナンバー,プルアップ(0:有効 1:無効)
↑ 必要であれば変更します。
- ◆WriteKM2IO20AkiCfg
↓
↓
WriteKM2IO20AkiCfg モジュールナンバー
- ◆再起動

I/Oモジュールを再起動
I/Oモジュールの電源を入れ直します。

CmdJigTool	KM2IO20Aki_IOBit.hsp		
	USB-IO2.0(AKI) IOBitドライバー		
	更新日時	20240219	(c)2023-2024 team.Drakuji

- コマンドジグについて
- KM2IO20Aki_CmdJigTool.hsp
- このプログラムは、I/Oモジュールの動作確認を簡易に行うジグです。

●項目説明

カテゴリー	機能	値	説明
▼MOD:モジュール			
	接続数		
	接続数を選択します。		
		0 自動	
		1から16	手動接続数選択
	接続		
	I/Oモジュールを接続し、IOBitを初期化します。		
	切断		
	I/Oモジュールを切断し、IOBitを停止します。		
▼I/O:入出力			
	<MOD#>		
	モジュールナンバーを選択します。		
	※以下全ての機能は、選択されたモジュールナンバーのI/Oモジュールに対して制御します。		
		0から15	モジュールナンバー(接続されている数に応じる)
	ADR		
	モジュールアドレスが有効な場合、モジュールアドレスを表示します。		
		-1	モジュールアドレス無効または取得失敗
		0から15	モジュールアドレス
	同期		
	モジュールナンバーで選択中のモジュールのピン状態を同期します。		
	連続同期		
	同期を連続で行います。		
		チェック	チェック後に同期を押すと、連続同期
		未チェック	チェックを外すと同期を押すたびに同期
	<PIN>		
	ピン名を表示します。		
	<状態>		
	各ピンの状態を表示、チェックで指定します。		
		0	Loレベル
		1	Hiレベル
		チェック	同期後Hiレベル
		未チェック	同期後Loレベル
	<モード>		
	各ピンのピンモードを表示、チェックで指定します。		
		D_OUT	デジタル出力
		D_IN	デジタル入力
		A_OUT	疑似PWM出力
		チェック	設定適用後ピンモードが入力(D_IN)
		未チェック	設定適用後ピンモードが出力(D_OUT)
▼設定			
	プルアップ無効		
	J2ピン群のプルアップ機能を指定します。		

チェック	設定適用後プルアップが無効
未チェック	設定適用後プルアップが有効

適用

ピンモードおよびプルアップ設定をI/Oモジュールに適用します。

※適用後、I/Oモジュールの再起動が必要です。