

USB-IO Family

サンプルプログラム一覧

USB-I2C.0

フォルダ	プログラム	回路図
00_Setting	USB-I2C.0設定	
00_SettingFSIO	USB-FSIO設定	
00_SettingSerial	シリアル番号設定	
01_Input	入力	
02_InOut	出力	
05_LEDStand	LEDスタンド	05_LED_Stand. PNG
06_LED_12	12個のLED制御	06_12LED. PNG
07_100V	100V電源制御	07_SSR. PNG
08_LCD	LCD表示	08_LCD. PNG

USB-FSIO

フォルダ	プログラム	回路図
10_RC_Servo	RCサーボ制御	10_RC_SERVO. PNG
11_LEDStandPWM	LEDスタンドPWM制御	11_LED_PWM. PNG
12_5V	5V電圧計	12_5VPNG
13_LCDFirst	LCD表示高速化	13_LCD_First. PNG
14_3G	加速度センサ アナログ入力	14_3G. PNG
15_Weather	温度・湿度・気圧 加速度センサ アナログ入力	15_Weather1. PNG 15_Weather2. PNG

環境別設定

64Bit環境

サンプルプログラムをそのままご利用ください。

32Bit環境

サンプルプログラムの変更が必要です。

1. ソリューションエクスプローラーからUSB-I0_Family.dllを削除
2. ソリューションエクスプローラーでソリューション名を右クリックし、追加－既存の項目をクリックしてから¥SRC¥DLL¥x86¥USB-I0_Family32.dllを追加
3. ソリューションエクスプローラーでソリューション名を右クリックし、参照の追加をクリックして、参照タブのUSB-I0_Family32.dllを選択
4. ソリューションエクスプローラーでUSB-I0_Family32.dllをクリックし、出力ディレクトリにコピーを常にコピーに変更
5. ツール－オプションのプロジェクトおよびソリューションのビルド構成の詳細を表示にチェック
6. ビルド－構成マネージャをクリックし、アクティブソリューションプラットフォームを新規作成でx86を作成しプラットフォームをx86に変更
7. Form1.vbのソースのImportsを USB_I0_Family32に変更

USB-I/O Family設定

デバイス接続後、USB-I/O2.0の場合は、 00_Setting USB-FSIO
の場合は、 00_SettingFSIO を起動し、以下の設定後を抜き差しして設定を完了してください。

プログラム	設定
01_Input	入力ピン設定 ポート1 11111111
	入力ピン設定 ポート2 1111
	内臓プルアップ無効 チェックなし
02_InOut	入力ピン設定 ポート1 00000000
	入力ピン設定 ポート2 1111
	内臓プルアップ無効 チェックなし
05_LEDStand	入力ピン設定 ポート1 00000000
	入力ピン設定 ポート2 1111
	内臓プルアップ無効 チェックなし
06_LED_12	入力ピン設定 ポート1 00000000
	入力ピン設定 ポート2 0000
	内臓プルアップ無効 チェック
07_100V	入力ピン設定 ポート1 00000000
	入力ピン設定 ポート2 1111
	内臓プルアップ無効 チェックなし
08_LCD	入力ピン設定 ポート1 00000000
	入力ピン設定 ポート2 0000
	内臓プルアップ無効 チェック

10_RC_Servo	アナログ入力数		8
	内臓プルアップ無効		ポート 2 無効
	入力ピン設定	ポート 1	11111111
	入力ピン設定	ポート 2	00000000
	入力ピン設定	ポート 3	00000000
	入力ピン設定	ポート 4	11111111
	出力初期設定	ポート 1	00000000
	出力初期設定	ポート 2	00000000
	出力初期設定	ポート 3	00000000
	出力初期設定	ポート 4	00000000
PWM		有効	
11_LEDStandPWM	アナログ入力数		8
	内臓プルアップ無効		ポート 2 無効
	入力ピン設定	ポート 1	11111111
	入力ピン設定	ポート 2	00000000
	入力ピン設定	ポート 3	00000000
	入力ピン設定	ポート 4	11111111
	出力初期設定	ポート 1	00000000
	出力初期設定	ポート 2	00000000
	出力初期設定	ポート 3	00000000
	出力初期設定	ポート 4	00000000
PWM		有効	

12_5V	アナログ入力数		8
	内臓プルアップ無効		ポート 2 無効
	入力ピン設定	ポート 1	11111111
	入力ピン設定	ポート 2	00000000
	入力ピン設定	ポート 3	00000000
	入力ピン設定	ポート 4	11111111
	出力初期設定	ポート 1	00000000
	出力初期設定	ポート 2	00000000
	出力初期設定	ポート 3	00000000
	出力初期設定	ポート 4	00000000
	PWM		有効
13_LCDFirst	アナログ入力数		0
	内臓プルアップ無効		ポート 2 無効
	入力ピン設定	ポート 1	00000000
	入力ピン設定	ポート 2	00000000
	入力ピン設定	ポート 3	00000000
	入力ピン設定	ポート 4	00000000
	出力初期設定	ポート 1	00000000
	出力初期設定	ポート 2	00000000
	出力初期設定	ポート 3	00000000
	出力初期設定	ポート 4	00000000
	PWM		有効

14_3G	アナログ入力数		8
	内臓プルアップ無効		ポート 2 無効
	入力ピン設定	ポート 1	11111111
	入力ピン設定	ポート 2	00000000
	入力ピン設定	ポート 3	00000000
	入力ピン設定	ポート 4	11111111
	出力初期設定	ポート 1	00000000
	出力初期設定	ポート 2	00000000
	出力初期設定	ポート 3	00000000
	出力初期設定	ポート 4	00000000
	PWM		有効
15_Weather	アナログ入力数		8
	内臓プルアップ無効		ポート 2, 4 無効
	入力ピン設定	ポート 1	11111111
	入力ピン設定	ポート 2	00000000
	入力ピン設定	ポート 3	00000000
	入力ピン設定	ポート 4	00000000
	出力初期設定	ポート 1	00000000
	出力初期設定	ポート 2	00000000
	出力初期設定	ポート 3	00000000
	出力初期設定	ポート 4	00000000
	PWM		有効

制御用DLLリファレンス

USB-I0_Familyクラス、USB-I0_Family32クラス 共通

構造体

設定内容構造体

ST_CTL_PROPERTY

Dim ADCount As Byte	アナログ入力数
Dim PullUpStop As Byte	プルアップ停止
Dim InputPort1 As Byte	入力ピン設定
Dim InputPort2 As Byte	入力ピン設定
Dim InputPort3 As Byte	入力ピン設定
Dim InputPort4 As Byte	入力ピン設定
Dim OutDefPort1 As Byte	出力ピン設定
Dim OutDefPort2 As Byte	出力ピン設定
Dim OutDefPort3 As Byte	出力ピン設定
Dim OutDefPort4 As Byte	出力ピン設定
Dim PwmEnable As Byte	PWM 有効無効

出力用構造体

ST_CTL_OUTPUT

Dim Port As Byte	出力ポート(1～4)
Dim Data As Byte	出力値

連続出力用構造体

ST_CTL_CONTINUE_OUTPUT

Dim Port As Byte	出力ポート(1～4)
Dim Data As Byte	出力値
Dim WaitL As Byte	833.3us単位の待ち時間
Dim WaitS As Byte	8.3us単位の待ち時間

入力用構造体

ST_CTL_INPUT

Dim Data() As Byte	入力値
※参考:Data(1)=ポート2 入力値	

アナログ入力用構造体		
ST_CTL_AD		
Dim Chanel As Byte		入力チャネル
Dim AD As Ushort		入力値
PWM間隔設定構造体		
ST_CTL_PWM_PERIOD		
Dim Period() As Ushort		66.83us単位の間隔
※参考: Period(1)=チャネル2 間隔値		
PWM出力設定構造体		
ST_CTL_PWM_DUTY		
Dim Duty() As Ushort		66.83us単位の出力時間
※参考: Duty(1)=チャネル2 出力値		

固定項目

項目	値	説明
MyVendorID	&h1352	Km2Net ベンダーID
MyProductIDs()	&H0110 &H0111 &H0120 &H0121	利用可能プロダクト ID USB-FSIO (KIT) USB-FSIO30 USB-IO2.0 USB-IO2.0(AKI)※ 秋月電子通商より販売
IO_PIN0	&H01	ピン番号 0
IO_PIN1	&H02	ピン番号 1
IO_PIN2	&H04	ピン番号 2
IO_PIN3	&H08	ピン番号 3
IO_PIN4	&H10	ピン番号 4
IO_PIN5	&H20	ピン番号 5
IO_PIN6	&H40	ピン番号 6
IO_PIN7	&H80	ピン番号 7
IO_PORT2PULLUP_STOP	&H01	ポート 2 プルアップ無効設定
IO_PORT4PULLUP_STOP	&H08	ポート 4 プルアップ無効設定
ERR_NONE	0	エラーなし
ERR_OPEN	-1	オープンエラー
ERR_NOT_OPEN	-2	オープンされていない
ERR_NOT_SUPPORT	-3	サポートされていない
ERR_OTHER	-9	その他エラー

参照値

項目	説明
gDeviceCnt As Integer	オープンデバイス数
gProductID() As UShort	オープン中プロダクト ID (0):最初に発見したデバイス (gDeviceCnt-1): 最大
gHidDevice() As Integer	オープン中デバイスハンドル (0):最初に発見したデバイス (gDeviceCnt-1): 最大

メソッド

メソッド	openDevice() As Integer
引数	なし
戻り値	0 以上:オープンデバイス数 0 未満:固定項目のエラー番号
説明 コンピュータに接続されている全ての USB-I0_Family をオープンします。	
メソッド	closeDevice()
引数	なし
戻り値	なし
説明 オープン中の全てのデバイスをクローズします。	
メソッド	getDeviceSerial() As Integer
引数	ByRef DeviceSerial As Byte Optional ByVal deviceNo As Integer = 0
戻り値	0:正常 0 未満:固定項目のエラー番号
説明 デバイス複数利用時にシリアル番号を取得します。 アプリケーションにてシリアル番号でデバイスを判断してください。	
メソッド	wrtDeviceSerial() As Integer
引数	ByVal DeviceSerial As Byte Optional ByVal deviceNo As Integer = 0
戻り値	0:正常 0 未満:固定項目のエラー番号
説明 デバイスを複数利用のために、シリアル番号をユーザーフラッシュメモリ領域の最後のバイトに書き込みます。	

メソッド	getProperty() As Integer
引数	ByRef prt As ST_CTL_PROPERTY Optional ByVal deviceNo As Integer = 0
戻り値	0:正常 0 未満:固定項目のエラー番号
説明 デバイスの設定内容を取得します	
メソッド	wrtProperty() As Integer
引数	ByVal prt As ST_CTL_PROPERTY Optional ByVal deviceNo As Integer = 0
戻り値	0:正常 0 未満:固定項目のエラー番号
説明 デバイスの設定内容を書き込みます。 書き込み後はデバイスを抜差してください。	
メソッド	ctlInOut() As Integer
引数	ByRef stInput As ST_CTL_INPUT ByVal stOutput() As ST_CTL_OUTPUT Optional ByVal deviceNo As Integer = 0
戻り値	0:正常 0 未満:固定項目のエラー番号
説明 デバイスから値を入力後、出力します。 出力したいポート数分、配列を定義しポートと値をセットしてください。	

メソッド	ctlOutIn() As Integer
引数	ByVal stOutput() As ST_CTL_OUTPUT ByRef stInput As ST_CTL_INPUT Optional ByVal deviceNo As Integer = 0
戻り値	0:正常 0 未満:固定項目のエラー番号
説明 デバイスから値を出力後、入力します。 出力したいポート数分、配列を定義しポートと値をセットしてください。	
メソッド	ctlADIn() As Integer
引数	ByRef stAD() As ST_CTL_AD Optional ByVal deviceNo As Integer = 0
戻り値	0:正常 0 未満:固定項目のエラー番号
説明 10BitA/D 変換した値を取得します。 入力したいチャンネル数分、配列を定義しチャンネルをセットしてください。	
メソッド	ctlPwmPreriod() As Integer
引数	ByVal stPeriod As ST_CTL_PWM_PERIOD Optional ByVal deviceNo As Integer = 0
戻り値	0:正常 0 未満:固定項目のエラー番号
説明 PWM の間隔を設定します。 構造体内の変数を配列定義し値をセットしてください。 配列の 0 番目が PWM チャンネル 1 です。 設定値×66.83us が時間です。	

メソッド	ctlPwmDuty() As Integer
引数	ByVal stDuty As ST_CTL_PWM_DUTY Optional ByVal deviceNo As Integer = 0
戻り値	0:正常 0 未満:固定項目のエラー番号
説明 PWM の出力を設定します。 構造体内の変数を配列定義し値をセットしてください。 配列の 0 番目が PWM チャンネル 1 です。 設定値×66.83us が時間です。	
メソッド	ctlPwmOn() As Integer
引数	ByVal PwmOn As Byte Optional ByVal deviceNo As Integer = 0
戻り値	0:正常 0 未満:固定項目のエラー番号
説明 PWM を ON/OFF します Bit0 が ON の時、チャンネル 1 が出力されます。	
メソッド	ctlCntOut() As Integer
引数	ByVal cntOut() As ST_CTL_CONTINUE_OUTPUT Optional ByVal deviceNo As Integer = 0
戻り値	0:正常 0 未満:固定項目のエラー番号
説明 連続出力を行います。 出力したい回数分、配列を定義しポートと値と時間をセットしてください。 構造体の中のWaitLは833.3us、WaitSは8.3us単位の待ち時間です。	