

IO 拓展模块使用说明书



版权声明

本手册版权归深圳市相诚质控技术有限公司所有，未经本公司书面许可，任何人不得翻印、翻译和抄袭本手册中的任何内容。

本手册中的信息资料仅供参考。由于改进设计和功能等原因，高川自动化保留对本资料的最终解释权，内容如有更改，恕不另行通知。



调试、运动中的机器有危险！使用者有责任在机器中设计有效的出错处理和安全保护机制，高川自动化没有义务或责任对由此造成的附带的或相应产生的损失负责。

目录

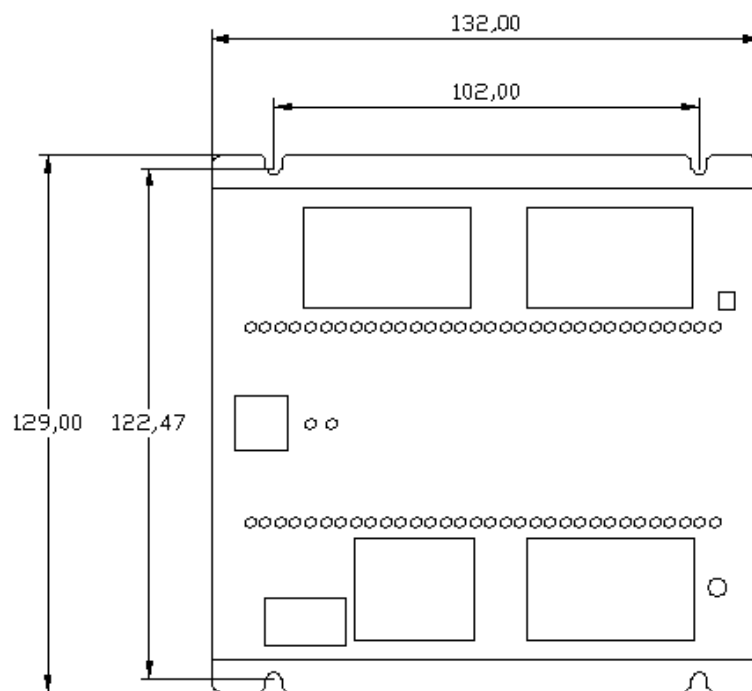
版权声明.....	- 2 -
联系方式.....	错误!未定义书签。
IO 拓展模块使用说明书	- 4 -
一. 性能规格.....	- 4 -
1. USBIO64M01 模块	- 4 -
2. USBIO32M01 模块	- 6 -
二. 接线说明.....	- 7 -
三. 模块测试.....	- 8 -
四. 软件编程.....	- 9 -
1. 挂接在 GCN 控制器上使用	- 9 -
2. 单独使用	- 11 -

IO 拓展模块使用说明书

一. 性能规格

USBIO64M01 和 USBIO32M01 提供一种低成本、简单易用、功能可靠的 IO 拓展方式，使用 RS485 或 USB2.0 接口与外部连接，可配合我司 GCN 系列控制器一起使用，也可单独使用，且无需安装驱动。

1. USBIO64M01 模块



规格 Specifications

隔离数字量输入	
通道	32
输入类型	干节点/湿节点
输入阻抗	5K Ω
隔离保护	2500 VDC
过压保护	70 VDC
ESD	2000 VDC
输入电流	3.5mA@0 VDC
输入电压	Logic 0: 4Vmax. Logic 1: 5V min(50V max)
隔离数字量输出	
通道	32
输出类型	Sink (NPN), 短路保护
输出电压	10~30VDC
隔离保护	2500 VDC
Sink 电流	100mA
通讯	
通讯接口	USB、RS485
常规规格	
尺寸	130.5X125mm
系统供电	24VDC +/- 20%
湿度	5 ~ 95% RH, non-condensing (IEC 68-2-3)
工作温度	0 ~ 60° C (32 ~ 140° F)
存储温度	-20 ~ 85° C (-4 ~ 185° F)
软件支持	
操作系统	Windows® XP/7/8 Linux WinCE
软件兼容性	VB/VC++/BCB/C#
演示软件	

2. USBIO32M01 模块

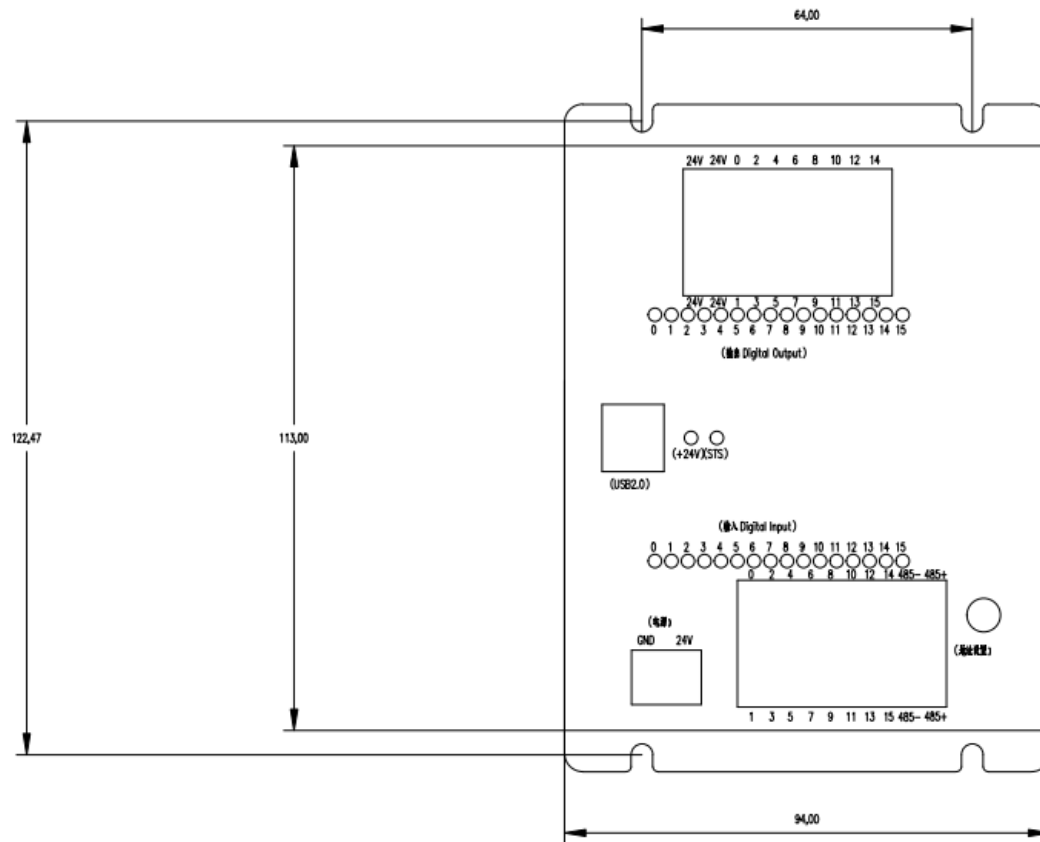


图 2.USBIO32M01 外观尺寸图

规格 Specifications

隔离数字量输入	
通道	16
输入类型	干节点/湿节点
输入阻抗	5K Ω
隔离保护	2500 VDC
过压保护	70 VDC
ESD	2000 VDC
输入电流	3.5mA@0 VDC
输入电压	Logic 0: 4Vmax. Logic 1: 5V min(50V max)
隔离数字量输出	
通道	16
输出类型	Sink (NPN), 短路保护
输出电压	10~30VDC
隔离保护	2500 VDC
Sink 电流	100mA
通讯	
通讯接口	USB、RS485

常规规格	
尺寸	94X123mm
系统供电	24VDC +/- 20%
湿度	5 ~ 95% RH, non-condensing (IEC 68-2-3)
工作温度	0 ~ 60° C (32 ~ 140° F)
存储温度	-20 ~ 85° C (-4 ~ 185° F)
软件支持	
操作系统	Windows® XP/7/8 Linux WinCE
软件兼容性	VB/VC++/BCB/C#
演示软件	

二. 接线说明

接线示意图如下：



图 3.拓展模块接线示意图

	需要注意的是，当拓展模块与控制器配合使用时，模块上的拨码地址从 2 开始，一个控制器最多可挂接 8 个拓展模块。 只有当模块单独使用时，才不需理会拨码地址，并且一个模块需要一个 USB 线与之单独通信，模块之间用 RS485 连接无效。
---	--

三. 模块测试

步骤如下：

1. 硬件连接，上电
2. 启动 GCS，点击菜单‘功能’—‘扩展模块’，启动扩展模块测试模块，如下图：



3. 切换扩展 IO 地址到连接模块的地址，通讯状态变为绿色，表示扩展模块通讯 OK。
4. 开始输入输出测试



单独使用时无法通过上述程序测试，需要时可联系我司技术人员

四. 软件编程

1. 挂接在 GCN 控制器上使用

1)使能模块

[NMC_GetIOModuleSts\(HAND devHandle, unsigned long *sts\);](#)

功能：读取扩展 IO 模块的状态

参数：

名称	数据类型	输入/输出	描述
devHandle	HAND	输入	设备句柄
sts	long *	输出	io 模块状态，按位指示，对应 bit 为 1 模块在线，0 不在线

[NMC_SetIOModuleEn\(HAND devHandle, unsigned char chDevId\);](#)

功能：设置扩展 IO 模块有效

参数

名称	数据类型	输入/输出	描述
devHandle	HAND	输入	设备句柄
chDevId	unsigned char	输入	设备 ID，取值[2,9]

2)设置通用输出

[NMC_SetDOGroup\(HAND devHandle, long value, short groupID\)](#)

功能：设置通用输出(按通道,支持超过 32 位)

参数：

名称	数据类型	输入/输出	描述
devHandle	HAND	输入	设备句柄
groupID	short	输入	DO 组，取值范围[0,n],具体需要看控制器是否存在多组数字量输出
value	long	输入	按位指示数字 IO 输出电平。 默认情况下，1 表示高电平，0 表示低电平。

[NMC_SetDO \(HAND devHandle, short groupID, long value \)](#)

功能：设置通用输出(按通道,支持超过 32 位)

参数：

名称	数据类型	输入/输出	描述
devHandle	HAND	输入	设备句柄
groupID	short	输入	DO 组，取值范围[0,n],具体需要看控制器是否存在多组数字量输出
value	long	输入	按位指示数字 IO 输出电平。

			默认情况下，1 表示高电平，0 表示低电平。
--	--	--	------------------------

[NMC_SetDOBit\(HAND devHandle, unsigned char bitIndex, unsigned char value \)](#)

功能：按位设置通用输出

参数：

名称	数据类型	输入/输出	描述
devHandle	HAND	输入	设备句柄
bitIndex	unsigned char	输入	取值范围[0,n],位序号,前 64 位为本地的通用输出，大于 64 为扩展 DO
value	unsigned long	输入	设置输出电平 默认情况下，1 表示高电平，0 表示低电平。

[NMC_GetDOGroup\(HAND devHandle, long *pDoValue, short groupID\)](#)

功能：读取通用输出(按通道,支持超过 32 位),带默认 group

参数：

名称	数据类型	输入/输出	描述
devHandle	HAND	输入	设备句柄
groupID	short	输入	取值范围[0,n],位序号,前 64 位为本地的通用输出，大于 64 为扩展 DO
value	Long*	输出	返回输出电平 默认情况下，1 表示高电平，0 表示低电平。

[NMC_GetDO\(HAND devHandle, short groupID, long *pDoValue\)](#)

功能：读取通用输出(按通道,支持超过 32 位)

参数：

名称	数据类型	输入/输出	描述
devHandle	HAND	输入	设备句柄
groupID	short	输入	取值范围[0,n],位序号,前 64 位为本地的通用输出，大于 64 为扩展 DO
value	Long*	输出	返回输出电平 默认情况下，1 表示高电平，0 表示低电平。

3)读取通用输入

[NMC_GetDIGroup\(HAND devHandle, long *pInValue, short groupID\)](#)

功能：读通用输入(按通道,支持超过 32 位),带默认 group

参数：

名称	数据类型	输入/输出	描述
devHandle	HAND	输入	设备句柄
groupID	short	输入	DI 组，取值范围[0,n],0: 本地 DI31~DI0, 1: 本地 DI63~DI32，其他指扩展 IO 模

			块
inValue	long *	输出	按位指示返回通用数字量输入状态 1,表示高电平/开路, 0, 表示低电平/闭合

[NMC_GetDI \(HAND devHandle, short groupID, long *inValue \);](#)

功能：读取通用输入

参数：

名称	数据类型	输入/输出	描述
devHandle	HAND	输入	设备句柄
groupID	short	输入	DI 组, 取值范围[0,n],0: 本地 DI31~DI0, 1: 本地 DI63~DI32, 其他指扩展 IO 模块
inValue	long *	输出	按位指示返回通用数字量输入状态 1,表示高电平/开路, 0, 表示低电平/闭合

[NMC_GetDIBit\(HAND devHandle, short bitIndex, short *bitValue \)](#)

功能：按位读取通用输入

参数：

名称	数据类型	输入/输出	描述
devHandle	HAND	输入	设备句柄
bitIndex	short	输入	位序号 0~n
bitValue	short*	输出	返回通用数字量输入状态 1,表示高电平/开路, 0, 表示低电平/闭合

2. 单独使用

当需要单独使用拓展模块时，请联系我司技术人员提供对应的库文件和例程。下面罗列出基本使用函数。

[NIO_Search\(U16 *pDevNo, TDevInfo *pInfoList\);](#)

功能：模块搜寻

名称	数据类型	输入/输出	描述
pDevNo	U16*	输出	搜索到的模块数量
pInfoList	TDevInfo*	输出	模块信息，结构体请参考头文件

[NIO_OpenByID\(const B8 *idStr, PHAND pDevHandle\);](#)

功能：模块打开

名称	数据类型	输入/输出	描述
idStr	const B8 *	输入	模块 ID 字符串，通过搜索模块函数获得的结构体 TDevInfo 中获得

pDevHandle	PHAND	输出	模块句柄

[NIO_Close\(HAND devHandle\);](#)

功能: 模块关闭

名称	数据类型	输入/输出	描述
devHandle	HAND *	输入	模块句柄

[NIO_GetDO\(HAND devHandle, U16 portStart, U16 portCount, U16 *value\);](#)

功能: 获取设备点位输出状态

名称	数据类型	输入/输出	描述
devHandle	HAND	输入	模块句柄
portStart	U16	输入	起始端口号, 取值范围(0-1)
portCount	U16	输入	要获取的端口数量, 取值范围(1-2)
value	U16 *	输出	用于存储端口状态值缓冲(每个端口 16 个点位), 每个点位:1 代表触发; 0 代表未触发

[NIO_SetDO\(HAND devHandle, U16 portStart, U16 portCount, U16 *value\);](#)

功能: 设置设备点位输出状态

名称	数据类型	输入/输出	描述
devHandle	HAND	输入	模块句柄
portStart	U16	输入	起始端口号, 取值范围(0-1)
portCount	U16	输入	要设置的端口数量, 取值范围(1-2)
value	U16 *	输出	要设置的端口状态值缓冲(每个端口 16 个点位), 每个点位:1 代表触发; 0 代表不触发

[NIO_GetDOBit\(HAND devHandle, U16 channel, U16 *value\);](#)

功能: 获取设备单个点位输入状态

名称	数据类型	输入/输出	描述
devHandle	HAND	输入	模块句柄
channel	U16	输入	点位位置, 取值范围: 0-31
value	U16 *	输出	用于存储获取的点位状态, 1 代表触发; 0 代表未触发

[NIO_SetDOBit\(HAND devHandle, U16 channel, U16 value\);](#)

功能: 设置设备单个点位输出状态

名称	数据类型	输入/输出	描述
devHandle	HAND	输入	模块句柄
channel	U16	输入	点位位置, 取值范围: 0-31
value	U16 *	输出	用于存储获取的点位状态, 1 代表触发; 0 代表未触发

[NIO_GetDI\(HAND devHandle, U16 portStart, U16 portCount, U16 *value\);](#)

功能: 获取设备点位输入状态

名称	数据类型	输入/输出	描述
devHandle	HAND	输入	模块句柄
portStart	U16	输入	起始端口号, 取值范围(0-1)
portCount	U16	输入	要获取的端口数量, 取值范围(1-2)
value	U16 *	输出	用于存储端口状态值缓冲(每个端口 16 个点位), 每个点位:1 代表触发; 0 代表未触发

[NIO_GetDIBit\(HAND devHandle, U16 channel, U16 *value\);](#)

功能: 获取设备单个点位输入状态

名称	数据类型	输入/输出	描述
devHandle	HAND	输入	模块句柄
channel	U16	输入	点位位置, 取值范围: 0-31
value	U16 *	输出	用于存储获取的点位状态, 1 代表触发; 0 代表未触发

文件版本

版本	时间	说明
1.0	2019.04.29	初稿