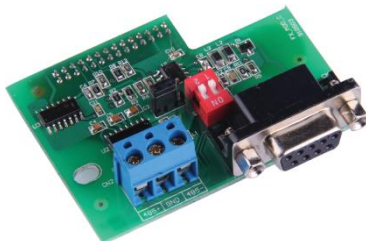




راهنمای نصب و راه اندازی
کارت مدباس

کارت مدباس

این مدل از کارت مدباس به وسیله ی اتصال از طریق دو نوع ارتباط (RS232 , RS485) انجام میشود. کارت مدباس به سوکت کارتهای آپشن برد کنترل وصل می شود.



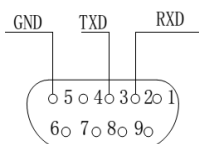
شکل (۱-۱) : کارت مدباس

دستورالعمل عملیاتی

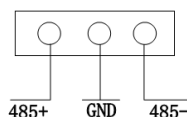
اگر نیاز به استفاده و کنترل اینورترهای سری VX با سیستم های کامپیوتری پیشرفته نظیر PLC و کامپیوتر های صنعتی را داشته باشید باید از این کارت ارتباطاتی استفاده نمایید. این کارت دو حالت ارتباطی فیزیکی را که همان RS232 و RS485 میباشد فراهم میسازد که میتوان متناسب با نوع ارتباط نیاز یکی را انتخاب نمود.

جک های ارتباطی

کارت مدباس دارای دو مدل کانکتور ارتباطی میباشد که در شکل ۱،۱ نمایش داده شده است.



D9: Bus-connector wiring terminal



RS485 wiring terminal

شکل (۱-۲) : نقشه جک های ارتباطی

اقدامات اولیه برای استفاده از کارت

- ۱) قبل از اتصال کارت به کنترل برد درایو حتما باید به صورت کامل خاموش گردد.
- ۲) کارت مدباس باید در اسلات کارتهای آپشن نصب گردد و از ارتباط کارت مدباس و برد کنترل مطمئن شوید.
- ۳) کارت مدباس با استفاده از پیچ در محل خود محکم شود.
- ۴) برای ارتباط مدباس از کابل دو رشته به هم تابیده شده شیلددار استفاده نمایید. همچنین کابل ارتباطی نباید در مسیر کابلهای قدرت قرار گیرد. تا نویزهای الکترومغناطیسی روی آن اثر نگذارند.
- ۵) بهتر است از زوج سیم های شیلد دار به جای ارتباط RS232 استفاده گردد.

پروتکل ارتباطی مدباس (Modbus)

اینورترهای سری VX پرت های ارتباطی RS232 و RS485 را برای ارتباط Master/Slave بر اساس پروتکل ارتباطی استاندارد مدباس فراهم میکنند.

کاربر میتواند برای پیاده سازی یک مرکز کنترل متمرکز از کامپیوتر یا PLC خود به منظور تنظیم دستورات کنترلی، فرکانس کاری، اصلاح پارامترها، وضعیت کاری اینورتر و نمایش پیغام های خطا استفاده نماید تا به نیازهای کاربردی خاص سیستم اینورترها برسد.

محصولات پروتکل

پروتکل ارتباطی سریال مدباس فریم و فرمت انتقال ناهنگام (asynchronous transmission) در ارتباط سریال را تعریف میکند که شامل دریافت و پخش فریم های Master و فرمت فریم های پاسخ Slave میباشد. محتوای فریم های Master شامل :

(1) آدرس پخش همگانی (Broadcast Address)

(2) دستورات اجرایی (Execution Command)

(3) داده (Data)

(4) بررسی خطا (Error Check)

(5) و غیره.....

پاسخ Slave نیز همان ساختار را دارد که شامل :

(1) تایید دریافت دستور (Action Confirmation)

(2) بازگردان داده ها (Data Return)

(3) بررسی خطا (Error Check)

(4) و غیره.....

در صورتی که خطایی به هنگام دریافت اطلاعات از Master توسط Slave صورت گیرد و یا داده به طور کامل انتقال پیدا نکند، Slave یک فریم خطا ایجاد میکند و برای Master ارسال کرده و منتظر پاسخ آن میماند.

حالت برنامه

اینورتر های VX فقط میتوانند به صورت "Single-Master Multi-Slave" کنترل شبکه را از طریق گذرگاه ارتباطی RS232/RS484 به دست بگیرند

ساختار گذرگاه (Bus Structure)

(۱) واسط سخت افزاری:

دو رابط سخت افزاری RS232/RS484

(۲) مد انتقال:

سریال ناهنگام (asynchronous serial) و حالت انتقال نیمه دو طرفه (half-duplex transmission) حالت های انتقال این پروتکل میباشد. در هر لحظه تنها یک Master و یک Slave میتوانند با هم تبادل دیتا داشته باشند در حالی که دیگر سیستم های متصل به گذرگاه فقط دیتا دریافت میکنند و در طول زمان این اتصال، داده ها به صورت فریم فریم به فرم بسته ارسال میگردند.

(۳) توپولوژی سیستم:

همانگونه که گفته شد توپولوژی ایت اتصالات به صورت "Single-Master Multi-Slave" میباشد. آدرس Slave ها در بازه ی ۱~۲۴۷ میباشد و صفر "۰" آدرس پخش همگانی است. در این توپولوژی هر Slave یک آدرس منحصر به فرد را دریافت کرده و با همان آدرس در شبکه ارتباطی شناخته میشود.

شرح پروتکل

پروتکل ارتباطی اینورتر های VX یک پروتکل سریال ناهنگام Master/Slave میباشد. تنها یک دستگاه (Master) میتواند یک پروتکل که Query/Command نامیده میشود را در شبکه ایجاد کند. دیگر دستگاه ها (Slave) تنها میتوانند داده های مورد نیاز Query/Command و اقدامات مربوطه به آن را برای Master بررسی و انجام دهند.

در اینجا Master اشاره دارد به کامپیوتر، کنترل کننده های صنعتی و PLC ها، همچنین Slave به اینورتر ها و دستگاه های کنترلی دیگری که از این پروتکل استفاده میکنند.

Master میتواند به صورت مستقل با یک Slave ارتباط داشته باشد و یا به صورت همگانی برای همه ی Slave ها داده ها را ارسال کند. برای انواع مختلفی از Query/Command که از طرف Master ارسال میگردد در صورتی

که به صورت مستقل ارسال گردد، Slave وظیفه دارد که یک پیغام که Response نام دارد برگرداند ولی در صورتی که پیغام به صورت همگانی ارسال گردد Slave نیازی به جواب دادن ندارد.

ساختار فریم ارتباطی

فرمت داده ها انتقالی در این پروتکل RTU (Remote Terminal Unit) یا همان واحد ترمینالی از راه دور میباشد. در این روش فرمت هر بایت به شکل زیر است :

(۱) بیت شروع فریم بایت

(۲) هشت بیت باینری که میتوانند به دو کاراکتر هگزادسیمال تبدیل شوند

(۳) بیت بررسی توازن (Parity bit)

(۴) بیت پایان فریم بایت

Start bit	Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7	Bit 8	A: Odd Parity Check bit B: Even Parity Check bit C : No Parity Check bit	Stop Bit
-----------	-------	-------	-------	-------	-------	-------	-------	-------	--	----------

جدول (۱-۱) : بیت بایت فریم

در مد RTU ، فریمهای جدید همیشه در انتقال حداقل ۳,۵ بایت زمان انتظار در استارت دارند. در یک شبکه که از baud rate برای محاسبه سرعت انتقال استفاده می کند، زمان انتقال ۳,۵ بایت به سادگی قابل کنترل می باشد. متعاقباً فرمت داده های ارسالی نیز به ترتیب به صورت زیر است:

(۱) آدرس Slave

(۲) نوع تابع درخواستی که به صورت ارسال به Slave یا دریافت از Slave میباشد.

(۳) داده

(۴) کد افزونگی چرخشی CRC Code

نوع داده ی هر یک از قسمت ها بالا اعدادی بر مبنای هگزادسیمال میباشد.

تمام دستگاه هایی که درون یک شبکه قرار میگیرند اعم از درایو ها و سیستم های کامپیوتری همواره تمامی فعالیتهای گذرگاه ارتباطی را حتی زمانی که در حالتی با Master در حال تبادل داده نیستند یعنی در زمان تاخیر داخلی مشاهده میکنند.

هنگامی که اولین فیلد پیغام را که همان پیغام آدرس است دریافت میکنند، هر دستگاه درون شبکه دریافت بسته را تایید میکند و پس از دریافت آخرین بایت از بسته زمان مشابهی که همان ۳,۵ بایت که به منظور نشان دادن اتمام فریم در نظر گرفته شده بود، سپری میشود و پس از آن مجدداً یک انتقال فریم دیگر ایجاد میشود.

اطلاعات یک فریم باید بصورت رشته دیتاهای پی در پی انتقال داده شود. اگر یک فاصله ۱,۵ بایتی قبل از کامل شدن انتقال یک فریم کامل وجود داشته باشد، دستگاه دریافت کننده اطلاعات ناتمام را پاک خواهد کرد. و آخرین بایت را به اشتباه به عنوان آدرس فریم بعدی در نظر خواهد گرفت. همچنین اگر فاصله بین فریم جدید و فریم قبلی کمتر از ۳,۵ بایت باشد، دستگاه دریافت کننده آنرا بخشی از فریم قبلی در نظر خواهد گرفت. با توجه به این اختلالات در فریم ها مقدار نهایی CRC اشتباه شده و همین موضوع منجر به عدم ارتباط میگردد. ساختار استاندارد فریم RTU به شکل زیر است.

Frame header (START)	Transmission time of 3.5 bytes
Slave address field (ADDR)	Communication address: 0~247 (decimal) ("0" stands for the broadcast address) ("248~255" stands for the Reserved Address And Master Address)
Function field (CMD)	Read slave parameters :03H Write slave parameters :06H
Data field DATA (0) ~ DATA (N-1)	This part is the main content of communications
CRC CHK lower bit	Detection value: CRC value (16BIT).
CRC CHK higher bit	
Frame tail (END)	Transmission time of 3.5 bytes

جدول (۲-۱) : فریم استاندارد RTU

ساختار کدهای فرمان و داده های انتقالی

ساختار کدهای دریافت و خواندن از روی درایو ها

عمل خواندن از روی درایو ها به منظور خواندن پارامتر های کنترلی درایو و خواندن وضعیت های جریان، ولتاژ و... میباشد. برای مثال برای درایوی که آدرس آن ۱ و میخواهیم از آدرس 4000H که آدرس یک پارامتر ذخیره شده در میکرو درایو میباشد، یک بایت دیتا را بخوانیم، فرمت داده ارسالی و دریافتی درایو به صورت زیر میشود.

دستور ارسالی توسط Master

START	Transmission time of 3.5 bytes
ADDR	01H
CMD	03H
Higher bits of start address	00H
Lower bits of start address	04H
Higher bits of data number	00H
Lower bits of data number	01H
CRC CHK lower bit	C5H
CRC CHK higher bit	CBH
END	Transmission time of 3.5 bytes

جدول (۳-۱) : فریم ارسالی Master برای خواندن از Slave

پیغام دریافتی از Slave

START	Transmission time of 3.5 bytes
ADDR	01H
CMD	03H
Higher bits of byte number	00H
Lower bits of byte number	04H
Higher bits of data address 0004H	00H
Lower bits of data address 0004H	00H
CRC CHK lower bit	04H
CRC CHK higher bit	0BH
END	Transmission time of 3.5 bytes

جدول (۴-۱) : فریم ارسالی Slave

همانگونه که مشاهده میشود Master پیغامی به صورت 010300040001H را به همراه CBC5 H.CRC ارسال میکند و درایو یا همان Slave نیز پیغام 010300040000H را به همراه 0B04 H.CRC برمیگرداند که نشان دهنده آن است که پارامتر به آدرس 4000H دارای مقدار 0000H میباشد.

ساختار کدهای انتقال داده و نوشتن بر روی درایو ها

عمل نوشتن بر روی درایو ها به منظور فرمان دادن به درایو ها و یا اصلاح و تغییر پارامتر های آنها میباشد. برای مثال برای درایوی که آدرس آن ۱ و میخواهیم آدرس 2000H که آدرس یک دستور کنترلی ذخیره شده در میکرو درایو میباشد را تغییر داده و مقدار 1388H را که معادل 5000D است و بیانگر این است که درایو در ۵۰٪ فرکانس نامی خود کار کند را تغییر دهیم. درایو یا همان Slave نیز پس از دریافت کامل داده ارسالی توسط Master دقیقاً همان دستور را برای Master ارسال میکند تا درستی فریم دریافتی بررسی گردد. فرمت داده ارسالی به درایو به صورت زیر میشود.

دستور ارسالی توسط Master

START	Transmission time of 3.5 bytes
ADDR	01H
CMD	06H
Higher bits of start address	20H
Lower bits of start address	00H
Higher bits of data number	13H
Lower bits of data number	88H
CRC CHK lower bit	8FH
CRC CHK higher bit	5CH
END	Transmission time of 3.5 bytes

جدول (۵-۱): فریم ارسالی Master برای نوشتن و یا فرمان دادن به Slave

پیغام دریافتی از Slave

START	Transmission time of 3.5 bytes
ADDR	01H
CMD	06H
Higher bits of start address	20H
Lower bits of start address	00H
Higher bits of data number	13H
Lower bits of data number	88H
CRC CHK lower bit	8FH
CRC CHK higher bit	5CH
END	Transmission time of 3.5 bytes

جدول (۶-۱): فریم ارسالی Slave در پاسخ Master

همانگونه که مشاهده میشود Master پیغامی به صورت 010620001388H را به همراه 5C8F H.CRC ارسال میکند و درایو یا همان Slave پس از دریافت آن و بررسی صحت آن فرکانس نامی خود را به ۵۰٪ میرساند.

بررسی خطا در فریم ارسالی

بررسی خطا در فریم های ارسالی شامل دو بخش میباشد

(۱) بایت بیت چک یا همان چک کردن بیت توازن (Parity Check)

(۲) چک کردن محتوای داخلی فریم به وسیله کد افزونگی چرخشی CRC Code

بیت توازن

بیتی است که برای نشان دادن زوج یا فرد بودن تعداد بیت هایی که ۱ می باشند به بیت ها اضافه می شود. بیت توازن در ساده ترین شکل برای مشخص کردن خطای کد به کار می رود. هنگامی که از توازن زوج استفاده می شود، اگر تعداد یکهای ورودی زوج باشد بیت توازن صفر می شود و بالعکس. و هنگامی که از توازن فرد استفاده می شود اگر تعداد یکهای ورودی فرد باشد بیت توازن صفر می شود و بالعکس.

۸ بیت خروجی همراه با بیت توازن		۷ بیت دریافتی
توازن زوج	توازن فرد	
10000000	00000000	0000000
10101001	00101001	0101001

جدول (۱-۶) : جدول بیت توازن Parity

بررسی افزونگی چرخشی (CRC)

فریم ارسالی به گذرگاه ارتباطی درایو ها و سیستم های کامپیوتری علاوه بر داده و آدرس های مورد نظر شامل فریمی به نام فریم شناسایی خطا (Error Detection Frame) هستند که بر اساس الگوریتم CRC محاسبه شده و به انتهای فریم اصلی اضافه میشود.

فریم CRC شامل دو بایت یا شانزده بیت باینری میباشد. پس از محاسبه ی آن توسط Master و ارسال آن به Slave، Slave این الگوریتم را بر روی داده های ارسالی اجرا میکند و حاصل محاسبه را با فریم CRC ارسال شده مقایسه میکند. در صورت مغایرت دو مقدار به این معناست که خطایی در داده های ارسالی رخ داده است. در این الگوریتم ابتدا مقدار هگزادسیمال FFFF در CRC قرار میگیرد سپس رشته ی ورودی بایت به بایت پردازش شده و مقدار CRC نهایی پس از پردازش بایت نهایی مشخص میگردد. هر بایت ورودی ابتدا با مقدار هگزادسیمال FFFF، XOR منطقی میشود و درون CRC ریخته میشود، سپس به تعداد ۸ بار یعنی تعداد ۱ بایت بر روی مقدار CRC عملیات زیر اجرا میشود.

(۱) مقدار CRC را شیفت به راست داده و با مقدار هگزادسیمال ۷.FFF، And منطقی کرده.

(۲) در صورتی که مقدار شیفت داده شده CRC بر ۲ باقیمانده ۱ داشته باشد CRC را با مقدار هگزادسیمال A001 ، XOR منطقی کرده در غیر اینصورت همان مقدار حاصل از And منطقی به مرحله بعدی منتقل میشود. چون فریم ارسالی در پروتکل مدباس ۱۶ بایتی است از الگوریتم CRC-16 Modbus استفاده میکنیم.

تعریف آدرس دیتای ارتباطی

در این قسمت تعریف آدرس دیتای ارتباطی مطرح می شود، که برای کنترل اپراتوری اینورتر استفاده می گردد و وضعیت اطلاعات و تنظیمات پارامترهای اینورتر را بدست می آوریم.

کد پارامترها

هر پارامتر یک شماره سریالی دارد که برای مشخص کردن آدرس رجیستر آن استفاده می شود که این شماره باید به هگزادسیمال تبدیل شود. برای مثال شماره سریال پارامتر P5.05 عدد ۸۲ می باشد. بنابراین آدرس آن بصورت هگزادسیمال 0052H خواهد بود که در قسمت آدرس فریم ارسالی برای اینورتر قرار میگیرد.

تنظیم پارامترهای ارتباط سریال درایو

PC.00	آدرس محلی درایو Address	مقدار دیفالت	1
	محدوده تنظیم	1-247 - آدرس ۰ به تمام اسلیوها پیغام ارسال می شود	

جدول (۷-۱): جدول آدرس Slave

هر کدام از درایوهای اسلیو باید یک آدرس اختصاصی داشته باشند. دو درایو اسلیو نمی توانند همزمان یک آدرس داشته باشند.

هنگامی که مستر پیغامی را در آدرس ۰ بفرستد، تمام اسلیوها آن پیغام را دریافت می کنند. ولی هیچکدام از اسلیوها به آن پاسخ نمی دهند.

PC.01	سرعت انتقال دیتا baud rate		مقدار دیفالت	3
	محدوده تنظیم		0	1200BPS
			1	2400 BPS
			2	4800 BPS
			3	9600 BPS
			4	19200 BPS
			5	38400 BPS

جدول (۸-۱): جدول سرعت انتقال دیتا

این پارامتر سرعت انتقال اطلاعات را بین مستر و اسلیوها مشخص می نماید. باید توجه داشت که مقدار baud rate در مستر و همه اسلیوها یکسان باشد. هر چه این پارامتر بالاتر باشد سرعت انتقال اطلاعات بیشتر خواهد بود.

0		مقدار دیفالت	فرمت داده Data format	PC.02
RTU : No parity: data format <8,N,1>		0	محدوده تنظیم	
RTU : Even parity: data format <8,E,1>		1		
RTU : Odd parity: data format <8,O,1>		2		
RTU : No parity: data format <8,N,2>		3		
RTU : Even parity: data format <8,E,2>		4		
RTU : Odd parity: data format <8,O,2>		5		
ASCII : No parity: data format <7,N,1>		6		
ASCII : Even parity: data format <7,E,1>		7		
ASCII : Odd parity: data format <7,O,1>		8		
ASCII : No parity: data format <7,N,2>		9		
ASCII : Even parity: data format <7,E,2>		10		
ASCII : Odd parity: data format <7,O,2>		11		

جدول (۹-۱): جدول فرمت داده بیت توازن

فرمت دیتا باید در مستر و اسلیوها یکسان باشد. در غیر اینصورت ارتباط برقرار نخواهد شد.

2ms	مقدار دیفالت	تاخیر در پاسخ time out	PC.03
0-20ms		محدوده تنظیم	

جدول (۱۰-۱): جدول تاخیر پاسخ

تاخیر پاسخ: فاصله زمانی بین دریافت اطلاعات توسط درایو و ارسال پاسخ به مستر می باشد. اگر این تاخیر کمتر از زمان پردازش اطلاعات باشد، آنرا به اندازه زمان پردازش اطلاعات افزایش دهید. و اگر این تاخیر بیشتر از زمان پردازش اطلاعات باشد. درایو تا زمان سپری شدن این تاخیر منتظر می ماند و سپس به مستر پاسخ می فرستد.

0.0 S	مقدار دیفالت	زمان انتظار فالت ارتباط سریال	PC.04
0.0 S غیر فعال 0.1 ~ 100.0 S		محدوده تنظیم	

جدول (۱۱-۱): جدول زمان انتظار خطا

اگر مقدار پارامتر فوق 0 تنظیم شود، زمان تاخیر فالت ارتباط سریال غیر فعال می شود. هنگامیکه مقدار پارامتر بیش از 0 تنظیم شود. اگر فاصله بین ارتباط فعلی و ارتباط بعدی بیش از مقدار زمان تاخیر تنظیم شده باشد، درایو فالت ارتباط سریال خواهد داد (Err18) معمولاً پارامتر فوق غیر فعال تنظیم می شود.

0	مقدار دیفالت	حالت پاسخ Response action	PC.05
	0 : فعال 1 : غیر فعال	محدوده تنظیم	

جدول (۱۲-۱) : جدول حالت های پاسخ

0	مقدار دیفالت	حالت خطا Fault action	PC.09
	0 : آلارم و استپ موتور 1 : بدون آلارم و ادامه کار موتور 2 : بدون آلارم و استپ موتور اگر منبع رفرنس با ارتباط سریال تنظیم می شود 3 : بدون آلارم و استپ اگر منبع رفرنس از هر جا باشد	محدوده تنظیم	

جدول (۱۳-۱) : حالت خطای اینورتر

Partosanat

VX Series Inverter

Modbus Card

www.partosanat.com

info@partosanat.com

Tel : +98 21 88662288

Fax : +98 21 88887809