آشنایی با پروتکل مدباس (ModBus) و راه اندازی آن

مقدمه

مدباس (ModBus) یکی از پروتکل های معروف در زمینه شبکه های صنعتی (Industrial Network) می باشد. این پروتکل که معمولا در بستر فیزیکی RS485 پیاده سازی می شود ، کاربردهای زیادی در انواع پروژه های صنعتی و سیستم های ساختمانی دارد. در این مقاله به بررسی این پروتکل و پیاده سازی آن می پردازیم.

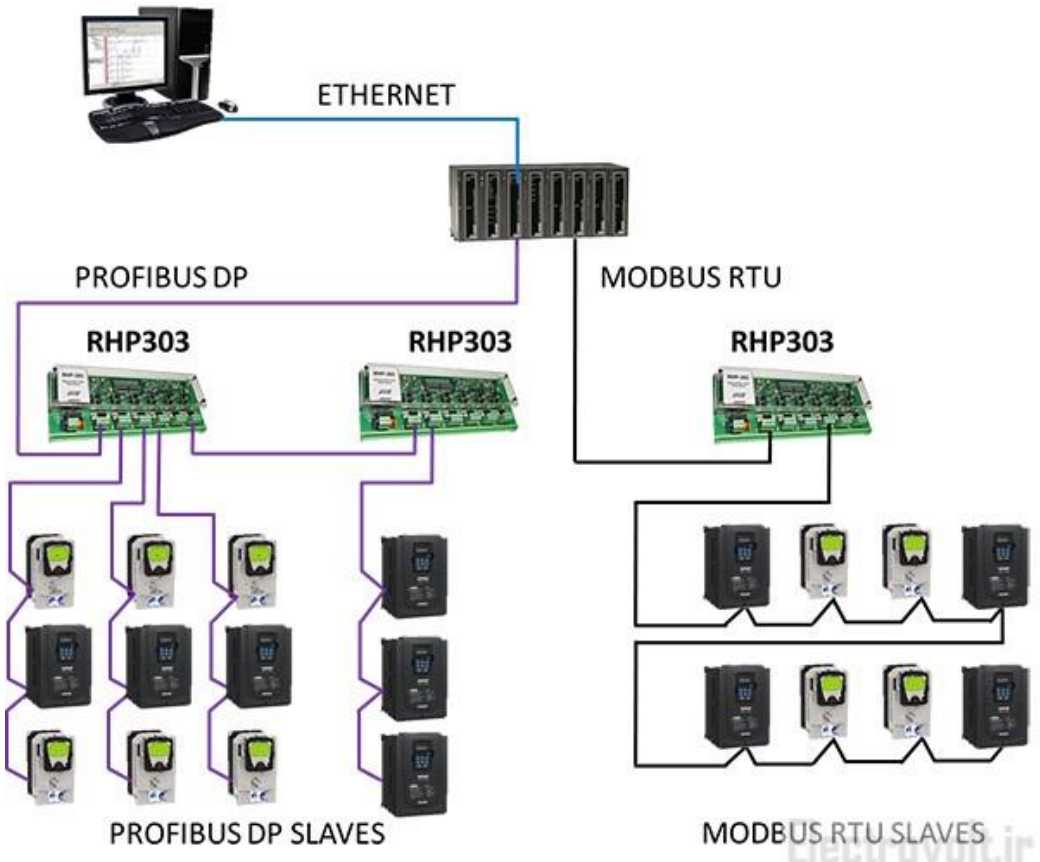
معرفی پروتکل مدباس

شبکه ModBus یک پروتکل ارتباطی است و ابتدا در سال 1979 توسط Modicon که امروزه Schneider Electric آن را در بر گرفته عرضه شد. کاربرد اولیه آن برای استفاده در PLC ها بود ولی بتدریج بعنوان یک استاندارد ارتباطی پذیرفته شد و بسیاری از سازندگان تجهیزات اتوماسیون آن را پشتیبانی کردند بدین ترتیب Modbus بصورت یک استاندارد باز در آمد بگونه ای که محصولات سازندگان مختلف به سهولت توسط این پروتکل با یکدیگر ارتباط برقرار کردند. سازندگان وسایل کوچک نیز ترجیح دادند این شبکه ModBus را با ارتباط RS232 و یا RS485 روی وسایل خود بکار ببرند تا استفاده از آنها در پروژه های بزرگ میسر گردد.

دلایلی که پروتکل Modbus در محیط های صنعتی کاربرد فراوانی دارد عبارتند از:

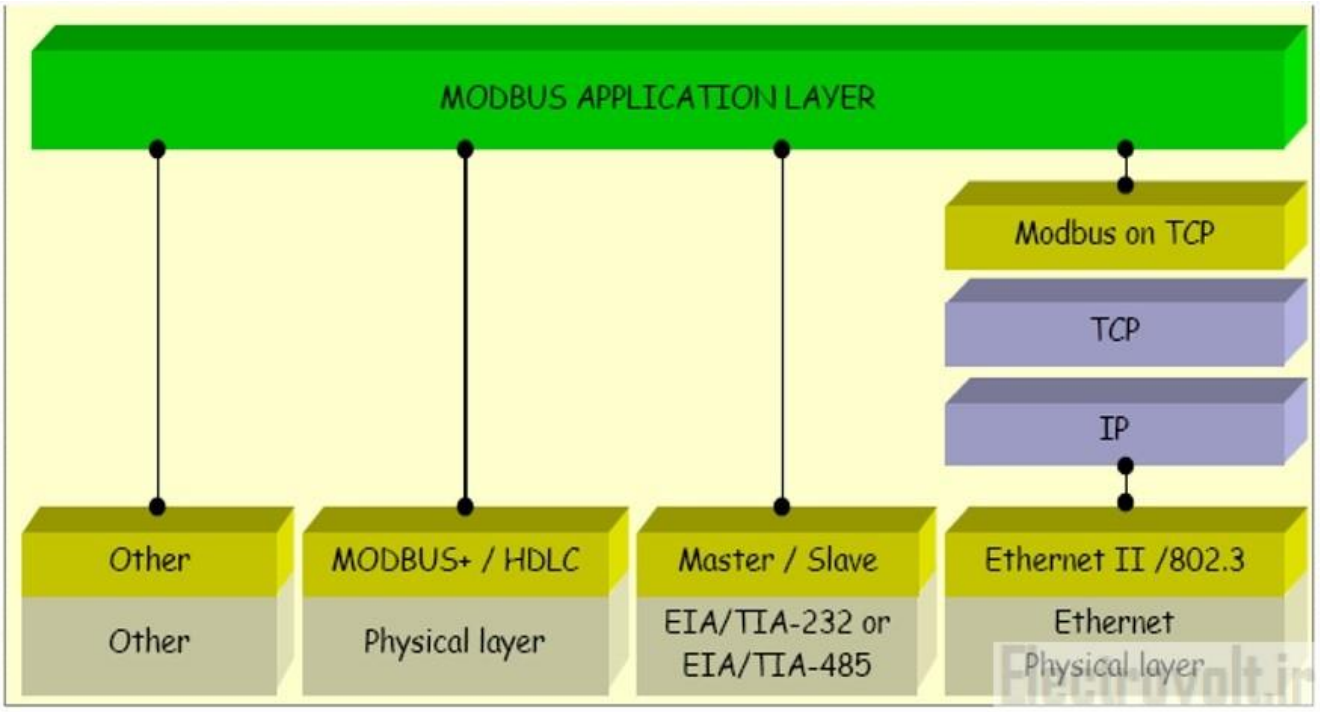
1. جهت کاربردهای صنعتی توسعه و گسترش یافته است
2. به صورت رایگان منتشر شده است
3. گسترش پذیری و نگهداری ساده ای دارد
4. اجازه انتقال بیت ها و بایت ها را بدون محدودیت خاصی به تولید کننده دستگاه می دهد

Modbusجهت برقراری ارتباط بین تعداد زیادی (تا 247) دستگاه متصل به یک شبکه استفاده می شود، به عنوان مثال دما و رطوبت اندازه گیری شده توسط سنسورها از طریق این پروتکل توسط کنترل کننده Master قرائت می شود. شکل زیر کاربرد این پروتکل در شبکه های صنعتی مبتنی بر PLC را نشان می دهد.



نگارش های پروتکل مدباس

- Modbus RTUمتداول ترین نگارش Modbus است. انتقال داده در این روش به صورت باینری فشرده صورت می گیرد.
- Modbus ASCIIاز کاراکترهای ASCII جهت انتقال داده استفاده می کند.
- Modbus/TCPاز استاندارد TCP/IP برای انتقال داده با سرعت بالاتر مورد استفاده قرار می گیرد.
- ModBus Plusکه بصورت Token Pass و با سرعت بالا طراحی شده است و یک باس انحصاری است.

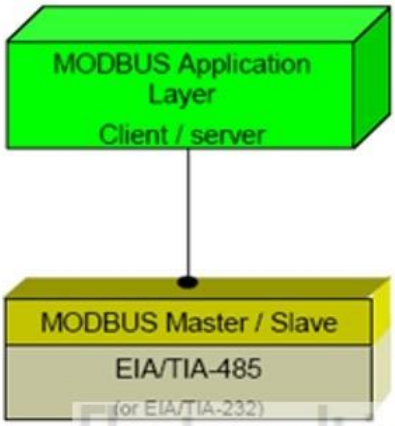


نحوه عملکرد پروتکل مدباس:

این پروتکل بصورت Master/Slave کار می کند. یعنی همواره یک وسیله بعنوان Master(فرمانده) و بقیه وسیله ها به عنوان Slave (فرمانبر) لحاظ می شوند. بدین صورت که هرگاه Master به یک دستگاه Slave با آدرس مخصوص خودش فرمانی ارسال می کند آن دستگاه جواب را به Masterبرمیگرداند. این پروتکل داده ها را از سطح فیلد دریافت و ضمن پردازش، آنها را به سطح نظارت ارسال می کند تا دستورالعمل کنترلی مناسب بر اساس داده های دریافتی، آلارم ها و رویدادها اتخاذ گردد MODBUS سریال از سرعت بالایی برخوردار بوده و بدون هر گونه Internal به تبادل اطلاعات می پردازد.

نکته: مد RTU که بعضا به آن ModBus-B بعنوان ModBus Binary گفته می شود مد اصلی است، مد ASCII که بعضا ModBus-A نیز گفته می شود برای برخی پیغام ها به کار می رود که این پیغامها طول شان دو برابر پیغام های RTU می باشد.
پروتکل مدباس از لایه های 2،1 و 7 مدل OSI استفاده می کند و در لایه فیزیکی RS232/RS485 را به کار می برد. شکل زیر این موضوع را نشان می دهد.

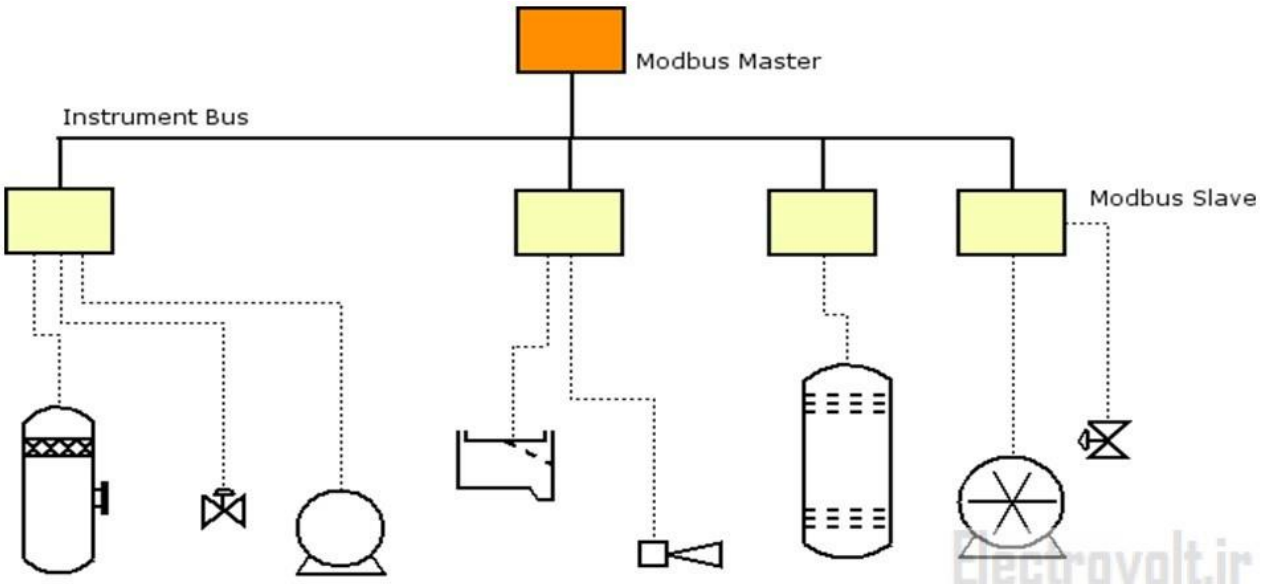
Layer	ISO/OSI Model	
7	Application	MODBUS Application Protocol
6	Presentation	Empty
5	Session	Empty
4	Transport	Empty
3	Network	Empty
2	Data Link	MODBUS Serial Line Protocol
1	Physical	EIA/TIA-485 (or EIA/TIA-232)



انواع حالت های عملکرد در پروتکل مدباس

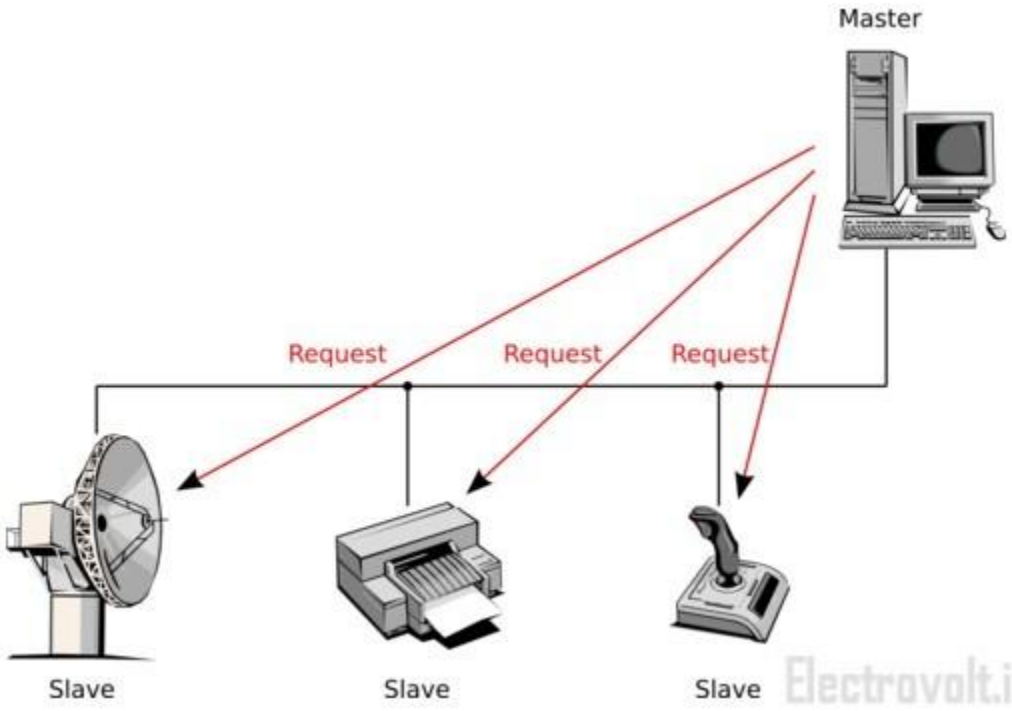
مدUnicast :

در این حالت Master از Slaveخاصی درخواست دیتا می نماید Slave.پس از دریافتRequest پیامReply را به Master ارسال می کند. بدیهی است هر Slaveباید دارای آدرس خاص و منحصربفردی باشد تا Master بتواند با آن ارتباط برقرار کند.

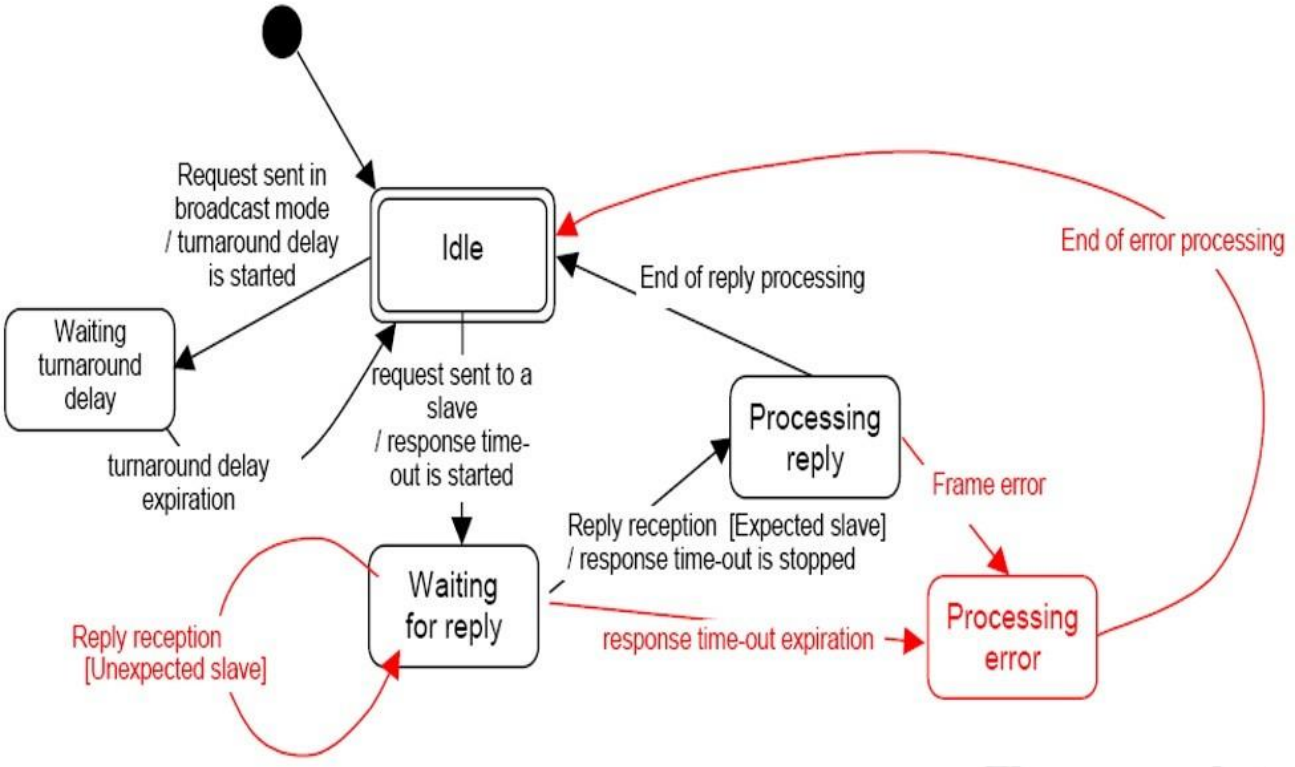


مد broadcast :

در این حالت Masterپیام خود را به تمام Slaveها میفرستد ولی هیچ پاسخی به Master بر نمیگردد. این مد از جمله برای نوشتن فرامین (Writing Commands) به کار می رود.

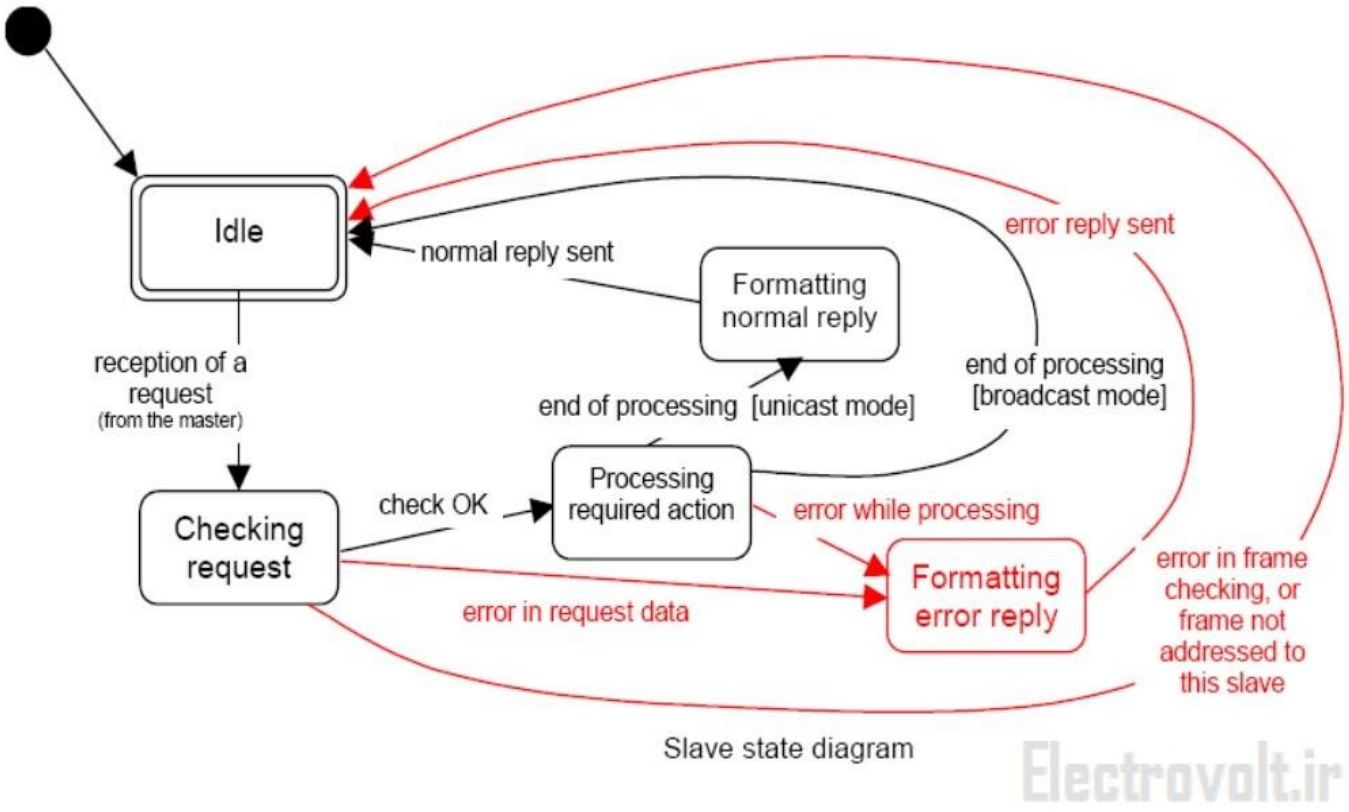


دیاگرام وضعیت Master :



Master state diagram

دیاگرام وضعیت: Slave

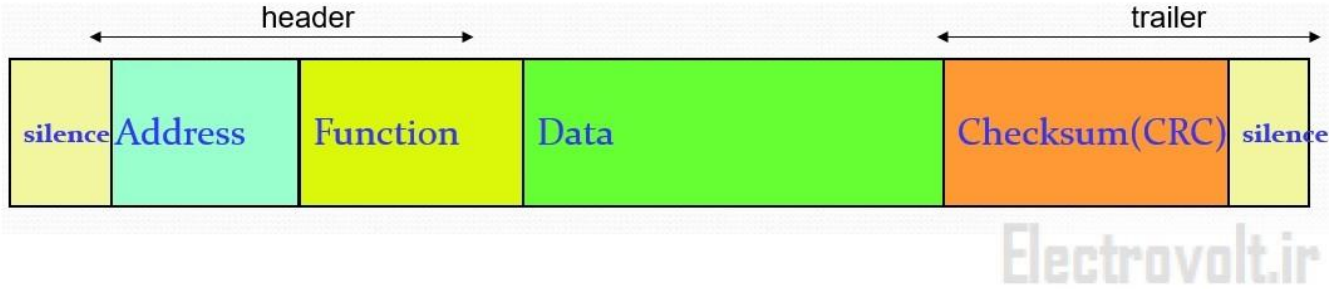


قالب استاندارد مدباس: RTU

در پروتکل مدباس RTU قالب استاندارد به صورت زیر است:

```
Modbus frame = Start + Address + Function code + Data + CRC Error check + End
```

که در شکل زیر هر یک از این بخش ها را مشاهده می کنید:



آدرس (Address) :

هرslave دارای یک شناسه از 1تا 247 می باشد .زمانی که master درخواست اطلاعات می کند اولین بایت پیغام به عنوان آدرس slave فرستاده می شود. بدین صورت slave متوجه می شود که پیام ارسال شده برای آن است یا نه. در واقع Master در ارتباط Unicast آدرس را در این فیلد قرار می دهد و Slave نیز وقتی پاسخ میدهد آدرس خودش را در این فیلد می گذارد تا Master بفهمد کدام Slave پاسخ داده است.

عملکرد (Function Code) :

دومین بایتی که توسط سیستم master ارسال می شود کد مربوط به انتخاب عملکردی است که باید انجام شود مثلا بایستی از Slave خوانده شود و یا در Slave نوشته شود.

دیتا (Data) :

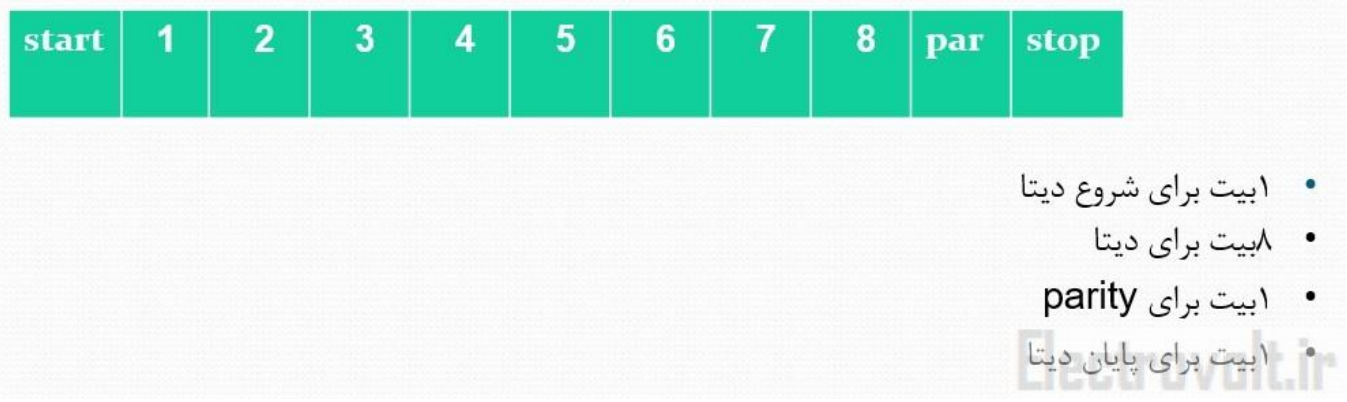
این فیلد حاوی اطلاعاتی بیشتر برای slave است تا عمل تعیین شده به وسیله ی function code ها را انجام دهد که بسته به نوع عملکرد انتخابی در مرحله قبل میتوند از 1 تا n بایت متغیر باشد.

بایت های چک (CRC) :

CRCدو بایت بوده و برای عیب یابی در انتهای پیغام آورده می شود.هر بایتی در پیغام برای محاسبه CRC استفاده می شود.دستگاه دریافت کننده نیز CRC را چک می نماید و با CRC موجود در پیغام master مقایسه می کند. اگر حتی یک بیت به درستی دریافت نشده باشد CRC ها متفاوت خواهند بود و باعث خطا می شوند.

بررسی مدباسRTU :

در این مد هر 8 بیت از بایت پیام شامل دو کاراکتر 4 بیتی هگزا دسیمال است. این ویژگی چگالی دیتا را افزایش داده و باعث می شود که نسبت به مد ASCII نرخ تبادل دیتا بهتر باشد. فرمت 11 بیت بسته دیتا در مد RTU به صورت زیر است:

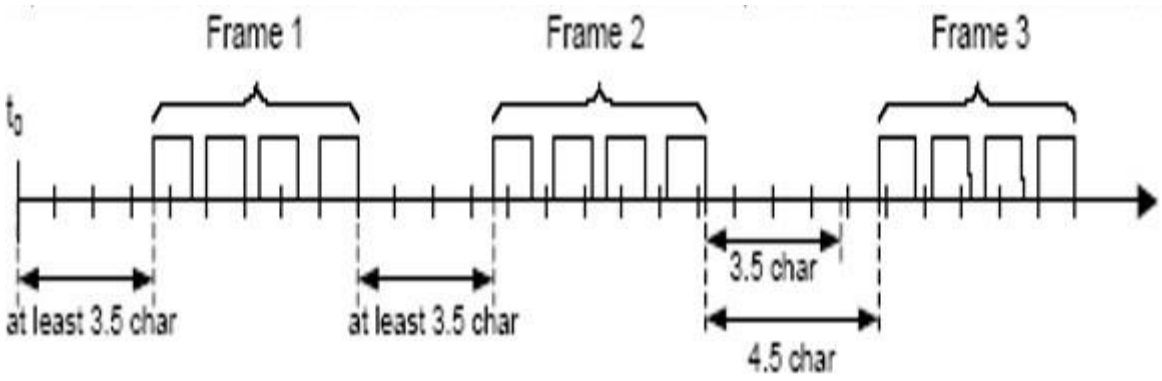


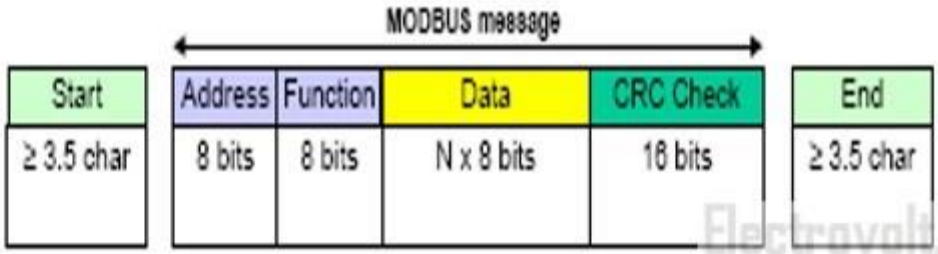
فرمت کلی در مدRTU :

Slave adress	Function code	data	CRC
1 byte	1 byte	0 up to 252 byte(s)	2 byte

ماکزیمم طول فریم در این مد ۲۵۶ بایت و ماکزیمم مقدار دیتا ۲۵۳ است.

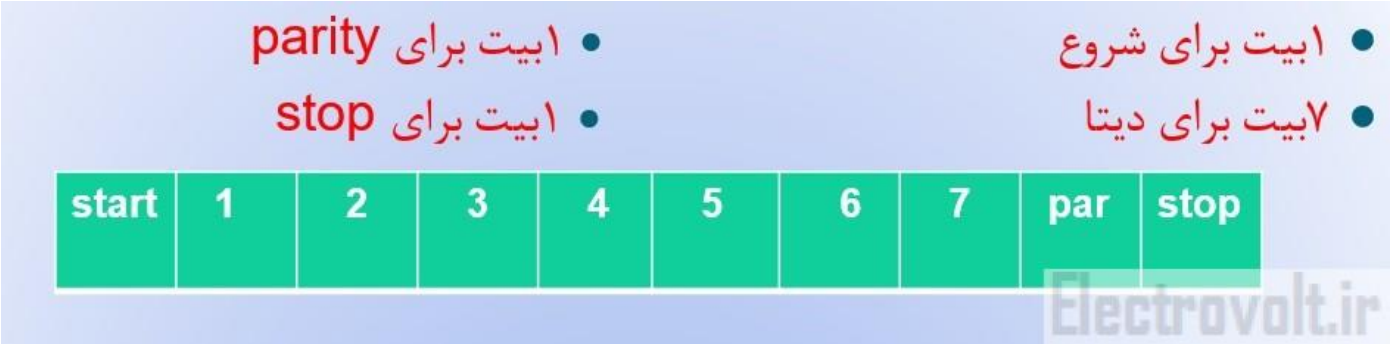
نکته: بین هر فریم یک فاصله زمانی وجود داردکه حداقل به اندازه 3.5 کاراکتر است و به آن فاصله خاموشی (silent Interval) نیزمیگویند. در نتیجه اگر بین دو کاراکتر متوالی یک فریم تاخیری بیش از 1.5 کاراکتر پیش بیاید نشان دهنده اشکال است. شکل زیر این موضوع را نشان می دهد.





بررسی مدباس ASCII :

در این مد هر 8 بیت از بایت پیام بصورت 2 کاراکتر ASCII ارسال می شود. از این رو بازدهی آن نسبت به RTU کمتر است. این مد در جایی که لینک فیزیکی یا قابلیت های وسیله ، اجازه استفاده از مد RTU را نمی دهد (به ویژه از نظر مدیریت تایمر ها) استفاده می شود. بعنوان مثال در این مد بایت 0 x5Bبه صورت دو بایت یعنی 0 x35=5و0 x42=Bدر می آید. در مد ASCII فرمت 10 بیت هر بسته دیتا به شکل زیر است:

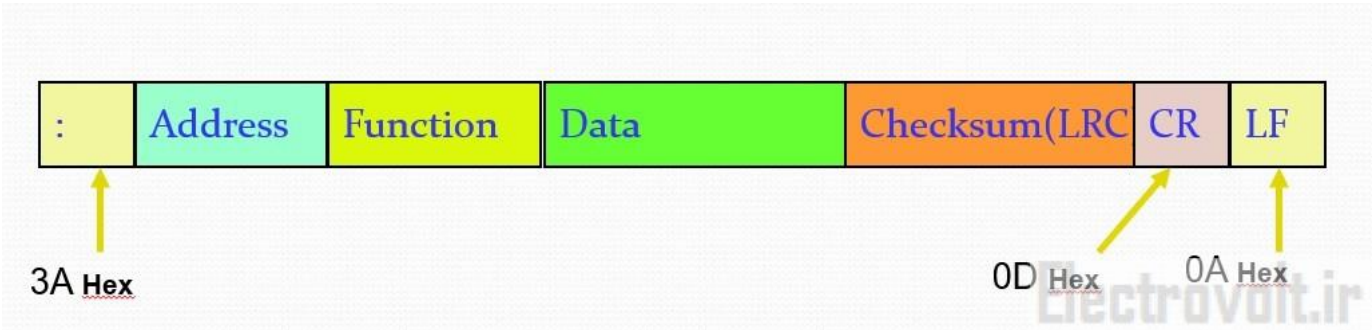


همچنین فریم پیام در مد ASCII به صورت شکل زیر است:

فریم پیام در مد ASCII:

Start	adress	Function	Data	LRC	END
1 char :	2 chars	2 chars	0 UP TO 2×252 char(s)	2 chars	2 chars CR,LF

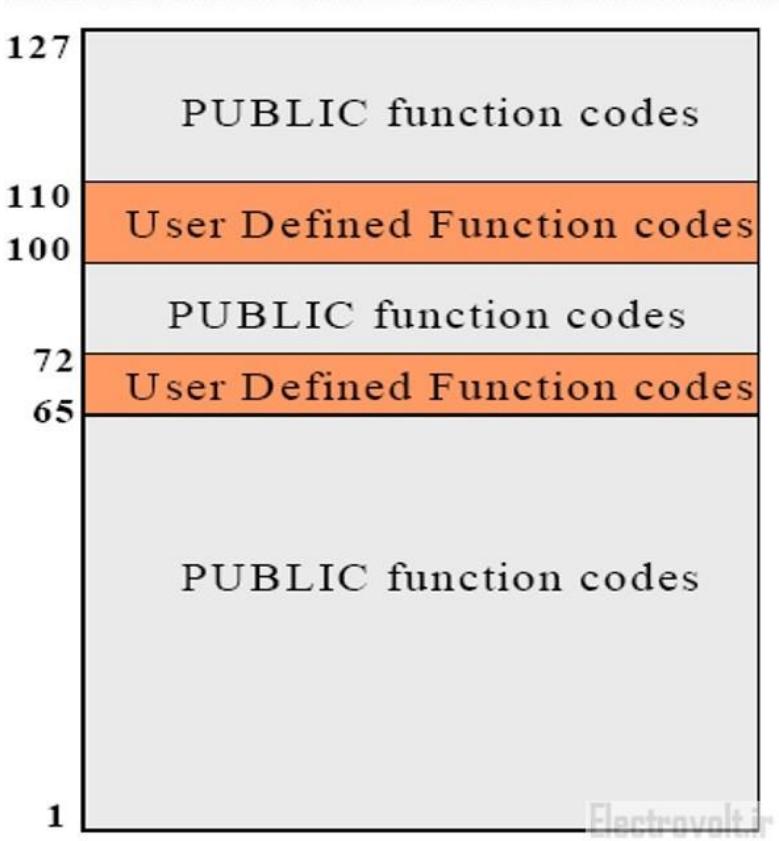
همانطور که مشاهده می کنید ، تجهیزات متصل به باس مرتباً باس را برای یافتن کاراکتر اول مانیتور می کنند. وقتی این کاراکتر دریافت شد وسیله کاراکتر بعدی را می گیرد که آدرس را مشخص میکند و اگر آدرس به او مربوط بود سایر کاراکترها را دریافت میکند تا بسته به پایان برسد. کاراکتر پایانیCarriage Return-Line Feed یا CRLF است. هر بایت دیتا نیاز به 2 کاراکتر برای کد کردن دیتا دارد بنابراین برای اطمینان از سازگاری بین مدهای ASCII و RTU ماکزیمم سایز اختصاص یافته 2×256 کاراکتر است یعنی دو برابرسایز دیتای . RTU بدین ترتیب سایز کل فریم مطابق شکل قبلی 513 کاراکتر خواهد بود.کنترل خطا در این مد توسط LRC انجام می شود.



فیلد Function Code یک بایت است که می تواند بین 1 تا 255 دسیمال باشد. کدهای 1 تا 127 برای کار نرمال و کدهای 128 تا 255 برای پاسخ های Exception که برای شرایط خطا طراحی شده بکار میرود. فیلد Data در برخی درخواست های خاص ممکن است خالی باشد (طول

صفر) این در مواردی است که ارسال Action به تنهایی برای Slave کفایت میکند و اطلاعات اضافی مورد نیاز نیست. وقتی خطایی وجود نداشته باشد Slave درخواست Master را انجام داده و در پاسخ خود به همان کد فانکشن اشاره میکندکه اصطلاحاً گفته میشود کد فانکشن اکو شده است. ولی وقتی اشکالی وجود داشته باشد و Slave نتواند عمل مورد درخواست Master را انجام دهد در این حالت پاسخی که در آن بجای کدفانکشن کد Exception آمده برگشت داده میشود تا Master از بروز خطا مطلع شود.

انواع کدهای عملکرد(Function Code)



1- عمومی: (Public)

بصورت استاندارد از قبل تعریف شده هستند و برای مقاصد مشخص بکار میروند و توسط modbus.org تایید شده هستند.

2- خاص: (User Define)

توسط کاربر تعریف می شوند و نیازی به تایید موسسه Modbus.org ندارند ولی باید توجه داشت که کد های رزرو شده را برای این فانکشن ها نمیتوان استفاده کرد. کد فانکشن های کاربر میتواند در محدوده 65 تا 72 یا 100 تا 110 باشد.

تشریح برخی از کدهای عملکرد:

- Code Function
- 01 (0x01) Read Coils
- 02 (0x02) Read Discrete Inputs
- 03 (0x03) Read Holding Registers
- 04 (0x04) Read Input Registers
- 05 (0x05) Write Single Coil
- 06 (0x06) Write Single Register
- 15 (0x0F) Write Multiple Coils
- 16 (0x10) Write Multiple Registers
- 23 (0x17) Read/Write Multiple Registers
- 43 (0x2B) Read Device Identification

- The complete description of all Modbus request is freely available on the Modbus.org web site :www.modbus.org

فانکشن کد 01 یا Read Coils :

این فانکشن برای خواندن وضعیت 1 تا 2000 خروجی از روی وسیله متصل به باس شبکه بکار میرود. در بسته PDU علاوه بر کد 01 آدرس اولین خروجی و تعداد آنها داده می شود. آدرس خروجی از صفر شروع می شود یعنی مثلاً خروجی های 1 تا 16 بصورت 0 تا 15 آدرس می گیرند. بنابراین PDU مربوط به درخواست بصورت شکل زیر خواهد بود:

Request

Function code	1 Byte	0x01
Starting Address	2 Bytes	0x0000 to 0xFFFF
Quantity of coils	2 Bytes	1 to 2000 (0x7D0)

در پاسخ وضعیت خروجی که 1 برای ON و 0 برای OFF است توسط بیت های دیتا مشخص می گردد. اگر تعداد خروجی ها مضربی از 8 نباشد در اینصورت در بایت آخر سایر بیتهای باقیمانده با صفر پر میشوند ولی از آنجا که تعداد خروجی ها نیز برگردانده می شود Master متوجه می شود که تا کجا مربوط به خروجی هاست.

Response

Function code	1 Byte	0x01
Byte count	1 Byte	N*
Coil Status	n Byte	n = N or N+1

*N= Quantity of Outputs / 8, if the remainder is different ofo N = N+1

اگر ERRORپیش بیاید کد ها بصورت زیر خواهند بود:

Function code	1 Byte	Function code + 0x80
Exception code	1 Byte	01 or 02 or 03 or 04

که کدهای استثنا یا Exception Code ها به صورت زیر می باشند:

Exception Codes

Code	Name	Meaning
01	ILLEGAL FUNCTION	The function code received in the query is not an allowable action for the slave. If a Poll Program Complete command was issued, this code indicates that no program function preceded it.
02	ILLEGAL DATA ADDRESS	The data address received in the query is not an allowable address for the slave.
03	ILLEGAL DATA VALUE	A value contained in the query data field is not an allowable value for the slave.
04	SLAVE DEVICE FAILURE	An unrecoverable error occurred while the slave was attempting to perform the requested action.
05	ACKNOWLEDGE	The slave has accepted the request and is processing it, but a long duration of time will be required to do so. This response is returned to prevent a timeout error from occurring in the master. The master can next issue a Poll Program Complete message to determine if processing is completed.
06	SLAVE DEVICE BUSY	The slave is engaged in processing a long–duration program command. The master should retransmit the message later when the slave is free.

محدودیت های شبکهModBus

Modbusبه دلیل استفاده از لینک های سریال RS232-RS485 دارای محدودیت های شد که به برخی از آنها اشاره میگردد:

- کند بودن خطوط سریال که بین ۹۶۰۰ تا ۱۱۵۰۰۰بیت در ثانیه کار میکنند یعنی در ماکزیمم حالت ۰٫۱۱۵ mbps که این سرعت در مقایسه با شبکه های ارتباطی امروزی که ۱۰۰ Mbpsیا حتی چند Gbps سرعت دارند پایین است.
- از آنجا که توسط RS232 فقط دو وسیله و توسط RS485 بین ۲۰ تا ۳۰ وسیله امکان ارتباط دارند از اینرو برای ارتباط دادن تعداد زیادی وسایل مثلاً ۵۰۰ وسیله نیاز به ارتباطات پیچیده درختی شکل است.
- ارتباط سریال مدباس بصورت MASTER/SLAVE است بدین معنی که روی باس فقط یک وسیله (MASTER) اجازه صحبت با SLAVEها را دارد.

با وجود این محدودیت ها شبکه ModBus در عرصه صنعت کاربرد و جایگاه بسیاری دارد. معمولا در تابلو هایی که از تعداد زیادی اینورتر استفاده میشود شبکه مدباس بین plc و اینورتر ها راه اندازی میکنند.

دانلود سورس راه اندازی پروتکل ModBUS به زبان های C# و C++ شامل:

- سورس ارتباط کامپیوتر با Slave بوسیله مدباس) به زبان(C#
- سورس C++ و هدرفایل پروتکل مدباس برای استفاده در انواع میکروکنترلرها

لینک خرید آنلاین سورس این پروژه