

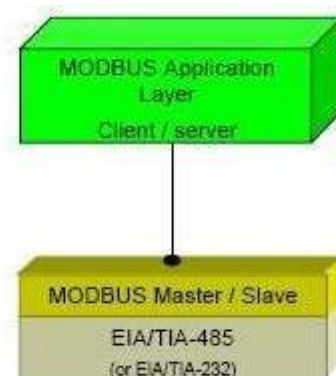
Modbus

Modbus ابتدا در سال ۱۹۷۹ توسط Modicon که امروزه Schneider Electric آنرا 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 پشتیبان کردند بدین ترتیب Modbus بصورت یک استاندارد باز در آمد بگونه ای که محصولات سازندگان مختلف بسهولت توسط این پروتکل با یکدیگر ارتباط برقرار کردند. سازندگان وسایل کوچک نیز ترجیح دادند این پروتکل را با ارتباط RS232 یا RS485 روی وسایل خود بکار ببرند تا استفاده از آنها در پروژه های بزرگ میسر گردد.

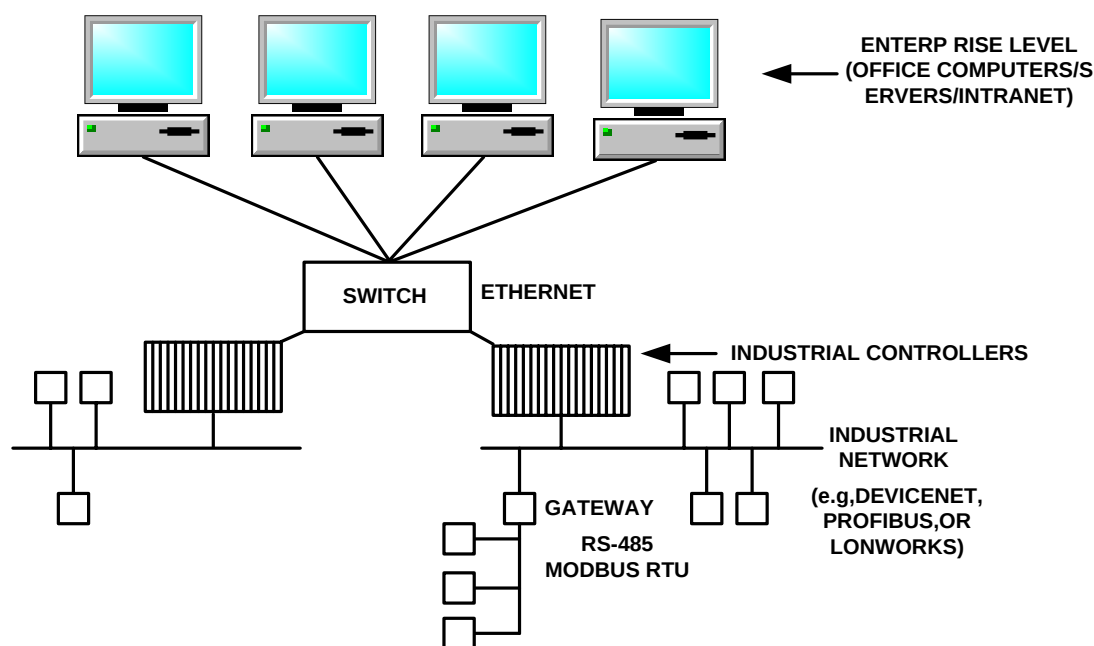
Modbus به دلیل استفاده از لینک های سریال RS232/RS485 دارای محدودیت های شد که به برخی از آنها اشاره میگرد:

- کند بودن خطوط سریال که بین ۹۶۰۰ تا ۱۱۵۰۰۰ بیت در ثانیه کار میکنند یعنی در ماکزیمم حالت 0.115Mbps که این سرعت در مقایسه با شبکه های ارتباطی امروزی که 100Mbps یا 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 ارتباط پیچیده درختی شکل است.
 - ارتباط سریال Modbus بصورت Master/Slave است بدین معنی که روی باس فقط یک وسیله (Master) اجازه صحبت با گروهی از Slave ها را دارد. (مد باس سیستم Multimaster را ساپورت نمی کند)
 - با وجود این محدودیت ها پروتکل Modbus در عرصه اتوماسیون جایگاه ویژه خود را پیدا کرد.
 - Modbus دارای سه نسخه اصلی زیر است که ویژگی های آنها با هم متفاوتند:
 - Modbus RTU/ASCII که بصورت سریال روی RS485 یا RS232 کار میکند.
 - Modbus TCP/IP که روی اترنت کار میکند.
 - Modbus Plus که بصورت Token Pass و با سرعت بالا طراحی شده است و یک باس انحصاری است.
- در بین پروتکل های Modbus آنچه را می توان در خانواده فیلد باس جای داد Modbus RTU/ASCII است. این پروتکل که نسخه پایه است از دو مد انتقال ASCII و RTU استفاده می کند.
- ASCII قابل خواندن که بعنوان مثال برای تست بکار میرود (فرمت ASCII)
 - RTU فشرده و سریع که برای کار نرمال بکار میرود. (فرمت هگزاسیمال)
- مد RTU که بعضاً به آن Modbus-B بعنوان Modbus Binary گفته میشود مد اصلی است. مد ASCII که بعضاً به آن Modbus-A نیز گفته میشود برای برخی پیغام ها بکار میرود این پیغام ها طول شان دو برابر پیغام های RTU می باشد.

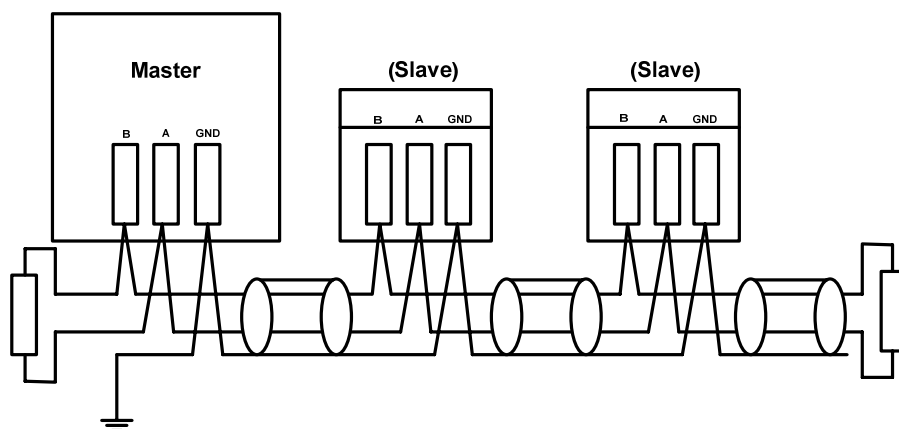
Layer	ISO/OSI Model	
7	Application	MODBUS Application Protocol
6	Presentation	Empty
5	Session	Empty
4	Transport	Empty
3	Network	Empty
2	Data Link	MODBUS Serial Line Protocol
1	Physical	EIA/TIA-485 (or EIA/TIA-232)



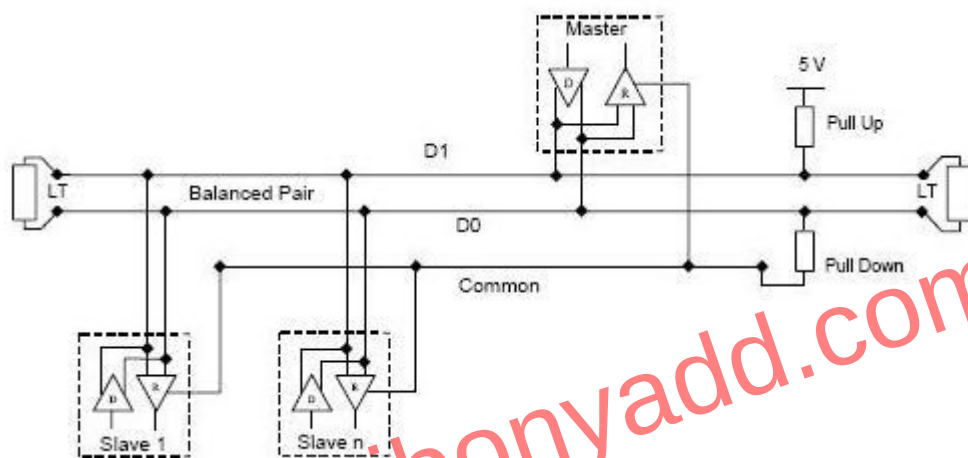
این پروتکل از لایه های ۱ و ۲ و ۷ مدل OSI استفاده میکند. در لایه فیزیکی RS232 و RS485 را بکار می برد. RS232 بصورت نقطه به نقطه P-t-P بسته میشود ولی RS485 علاوه بر P-t-P میتواند بصورت Multipoint نیز بسته شود یعنی چندین دستگاه روی یک باس قرار گیرند. در اینحالت باس معمولاً ۲ سیمه است اگر چه امکان اتصال ۴ سیمه نیز وجود دارد.



- در اتصال RS485 چه بصورت دو سیمه و چه بصورت ۴ سیمه در نظر داشتن نکات زیر ضروری است :
- بطور معمول بدون ریپیتر ماکزیمم ۳۲ وسیله را میتوان روی Modbus RS485 متصل نمود .
 - توپولوژی اتصال بصورت Bus است که میتواند Daisy Chain یا Tap and Drop باشد.
 - طول کابل Trunk محدود است این محدودیت بستگی به سرعت انتقال دیتا ، مشخصات کابل (از جمله خازن کابل) ، تعداد Node های متصل شده بصورت Daisy Chain و نیز ۲ سیمه یا ۴ سیمه بودن سیستم دارد . برای سرعت 9600 bps با کابل AWG26 یا ضخیم تر و سیستم ۲ سیمه این طول ۱۰۰۰ متر است . برای سیستم ۴ سیمه با مشخصات فوق این طول به نصف کاهش می یابد .
 - طول کابل انشعابی کوتاه باشد و از ۲۰ متر تجاوز نکند . کلاً اگر n انشعاب وجود داشته باشد برای بدست آوردن ماکزیمم طول هر انشعاب عدد ۴۰ متر را بر n تقسیم میکنیم .
 - سیم مشترک (GND یا Common) به زمین (PE) متصل شود ترجیحا در یکطرف و آنهم در سمت Master



- دو طرف کابل Trunk یعنی باس اصلی توسط ترمینیتور بسته شود . هر زوج سیم دیتا در دو طرف ترمینیتور لازم دارد پس در سیستم ۴ سیمه در هر طرف ۲ ترمینیتور لازم است .
- ترمینیتور میتواند یک مقاومت ۱۵۰ اهمی باشد . در برخی کاربردها از مقاومت ۱۲۰ اهمی که با خازن (10V minimum 1nF) سری شده استفاده می گردد .
- برای پلاریزه کردن خط از یک جفت مقاومت Pull Up و Pull Down که مقدار آنها می تواند بین ۴۵۰ تا ۶۵۰ اهم باشد در یکطرف (ترجیحاً سمت Master) استفاده میگردد . منظور از پلاریزه کردن خط مصون نگه داشتن باس از تداخل و نویز درحالتی است که روی خط هیچ دیتایی تبادل نمیشود لازم است در اینحالت گیرنده حالت ثابت خود را حفظ کند . مقاومت های فوق به تغذیه ۵ ولت متصل می شوند .
- ماکزیمم وسایل مجاز روی Modbus با پلاریزاسیون به یک چهارم حالت نرمال کاهش می یابد.



اتصال Modbus بصورت RS232

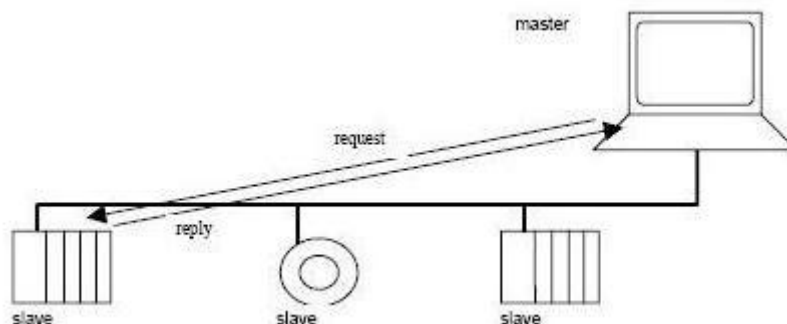
- این ارتباط بصورت نقطه به نقطه است یعنی فقط دو وسیله میتوانند با هم ارتباط داشته باشند .
- برای مسافت های کوتاه (معمولاً) کمتر از ۲۰ متر بکار می رود .
- ترمینیتور ندارد .

لایه دیتا لینک (لایه دوم)

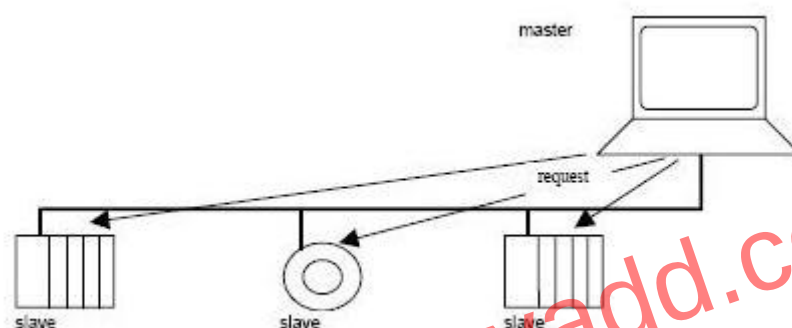
این لایه از تکنیک دسترسی Master/Slave استفاده میکند . بنابراین در هر لحظه فقط یک Master میتواند باس را در اختیار داشته و با ماکزیمم ۲۴۷ وسیله Slave که به همان باس متصل هستند ارتباط برقرار کند . بدیهی است در این تکنیک دسترسی Slave ها هیچگاه بدون درخواست Master دیتایی را نمیفرستند و هیچگاه نیز با یکدیگر ارتباط برقرار نمی کنند .

Master به یکی از دو روش زیر درخواست خود را ارسال می نماید :

۱ . مد Unicast : در این حالت Master از Slave خاصی درخواست دیتا می نماید . Slave پس از دریافت Request پیام reply را به Master می فرستد . بدیهی است هر Slave باید دارای آدرس خاص و منحصر بفردی باشد تا Master بتواند با آن ارتباط برقرار کند .



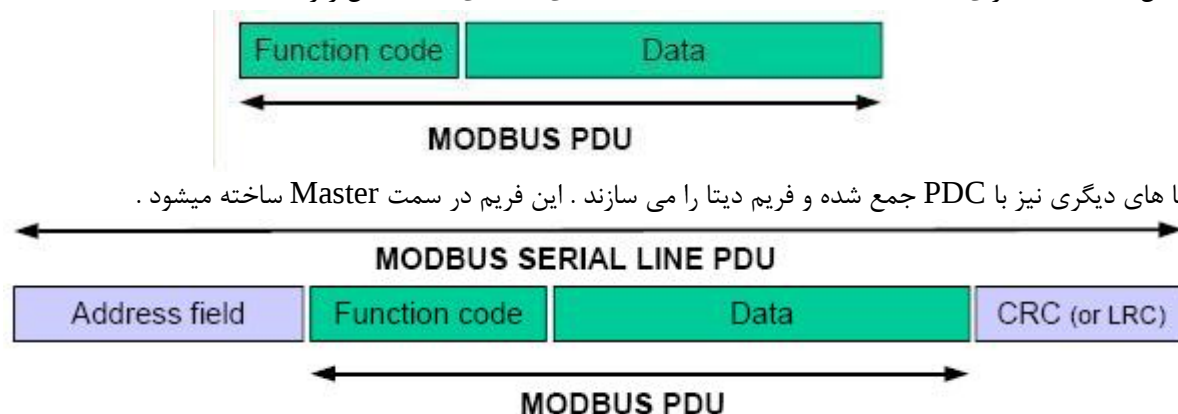
۲. مد Broadcast: در این حالت Master پیام خود را به تمام Slave ها میفرستد ولی هیچ پاسخی به Master بر نمیگردد. این مد از جمله برای نوشتن فرامین (Writing Commands) بکار میرود. برای مد Broadcast آدرس صفر رزرو شده است.



بطور کلی روش آدرس دهی در Modbus مانند جدول بعد است. باید توجه داشت که Master نیاز به هیچ آدرس خاصی ندارد و صرفاً Slave ها هستند که نیاز به آدرس دارند.

فریم اطلاعات

پروتکل Modbus دارای PDU (Protocol Data Unit) های ساده ای مانند شکل زیر است.



دیتا های دیگری نیز با PDC جمع شده و فریم دیتا را می سازند. این فریم در سمت Master ساخته میشود.

فیلد های فریم فوق بشرح ذیل هستند:

- فیلد آدرس صرفاً حاوی آدرس Slave است و میتواند بین ۱ تا ۲۴۷ باشد. Master در ارتباط Unicast آدرس را در فیلد قرار می دهد و Slave نیز وقتی پاسخ می دهد آدرس خودش را در فیلد می گذارد تا Master بفهمد کدام Slave پاسخ داده است.
- فیلد Function Code نشان دهنده عملی است که باید انجام شود (Action)

• فیلد CRC برای Error Checking است که تشریح خواهد شد .

مد های انتقال سریال

دو مد در انتقال سریال Modbus وجود دارد که عبارتند از : RTU و ASCII . این مدها وضعیت بیت های پیام و نحوه بسته بندی و باز شدن بسته پیام را مشخص میکند . تمام وسایل باید بتوانند مد RTU را که مد پیش فرض است بکار ببرند ولی مد ASCII اختیاری است .

مد RTU یا Remote Terminal

در این مد هر بیت ۸ بیت از بایت پیام شامل دو کاراکتر ۴ 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 . هر پیام بصورت رشته ای از کاراکترها ارسال میگردد . فرمت ۱۱ بیت بسته دیتا در مد RTU بصورت زیر است :

Start	1	2	3	4	5	6	7	8	Par	Stop
-------	---	---	---	---	---	---	---	---	-----	------

- ۱ بیت برای شروع دیتا
- ۸ بیت برای دیتا
- ۱ بیت برای Parity
- ۱ بیت برای پایان دیتا

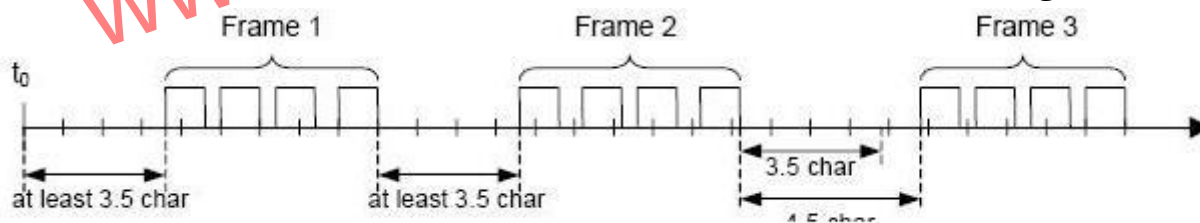
معمولا Parity زوج بصورت پیش فرض مورد استفاده قرار میگیرد اگرچه Parity فرد یا حتی بدون چک Parity نیز میتواند بکار رود . حالت اخیر برای اطمینان از ماکزیمم سازگاری با محصولات مختلف پیشنهاد میشود در این حالت ۲ بیت پایانی مورد نیاز است شکل زیر :

Start	1	2	3	4	5	6	7	8	Stop	Stop
-------	---	---	---	---	---	---	---	---	------	------

فرمت کلی فریم که شامل کاراکترهای مختلف است در مد RTU بصورت زیر است . همانطور که مشاهده میشود ماکزیمم طول فریم در این مد برابر ۲۵۶ بایت می باشد و ماکزیمم مقدار دیتا ۲۵۳ بایت است .

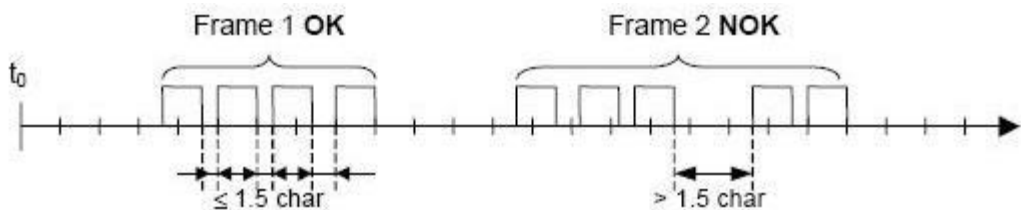
Slave Address	Function Code	Data	CRC
1 byte	1 byte	0 up to 252 byte(s)	2 bytes CRC Low, CRC Hi

بین فریم ها یک فاصله زمانی وجود دارد که حداقل به اندازه ۳.۵ کاراکتر است و به آن فاصله سکوت نیز میگویند .



MODBUS message				
Start	Address	Function	Data	CRC Check
≥ 3.5 char	8 bits	8 bits	N x 8 bits	16 bits
End				
≥ 3.5 char				

اگر بین دو کاراکتر متوالی یک فریم تاخیری بیش از ۱.۵ کاراکتر پیش بیاید نشان دهنده اشکال است .



این فواصل زمانی که به تایمرهای $t_{1.5}$ و $t_{3.5}$ نیز معروف است برای کنترل نیاز به وقفه های زیاد دارد که در سرعت های بالا بار زیادی را به CPU تحمیل میکند و بهتر است در سرعت های بالا زمان این تایمرها فیکس شوند. پیشنهاد میشود برای $t_{1.5}$ زمان ۷۵۰ میکرو ثانیه و برای $t_{3.5}$ زمان ۱۷۵۰ میلی ثانیه بکار برده شود.

مد انتقال ASCII یا American Standard for Information Interchange

در این مد هر ۸ بیت از بایت بصورت ۲ کاراکتر ASCII ارسال میشود از اینرو بازدهی آن نسبت به RTU کمتر است بعنوان مثال در این مد بایت OX5B بصورت ۲ بایت یعنی "5"=OX35 و "B"=OX42 در می آید.

یکی از محاسن مد ASCII فاصله زمانی بین دو فریم است که می تواند تا یک ثانیه باشد این امر باعث می شود که نیازی به سنکرون سازی ایستگاه های کاری وجود نداشته باشد. این مد در جایی که لینک فیزیکی یا قابلیت های وسیله اجازه استفاده از مد RTU را نمی دهد (بویژه از نظر مدیریت تایمرها) استفاده میگردد. در مد ASCII فرمت ۱۰ بیت هر بسته دیتا بصورت زیر است:

- ۱ بیت برای شروع
- ۷ بیت برای دیتا
- ۱ بیت برای Parity
- ۱ بیت برای Stop



نوع Parity شبیه آنچه برای RTU ذکر شد زوج بوده ولی انواع دیگر نیز قابل استفاده است. نحوه ارسال نیز شبیه RTU است یعنی از بیت LSB شروع و تا بیت MSB ادامه می یابد.

فریم پیامد در مد ASCII

هر فریم ASCII از بخشهایی مانند شکل زیر تشکیل شده است:

Start	Address	Function	Data	LRC	End
1 char	2 chars	2 chars	0 up to 2x252 char(s)	2 chars	2 chars CRLF

کاراکتر شروع یک کاراکتر که Colon (یعنی :) است و هگز آن 3A است می باشد. تجهیزات متصل به باس مرتباً باس را برای یافتن این کاراکتر 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 پایانی Carriage Return-Line Feed یا CRLF است که کد اسکی آن 0D و کد هگز آن 0A میباشد.

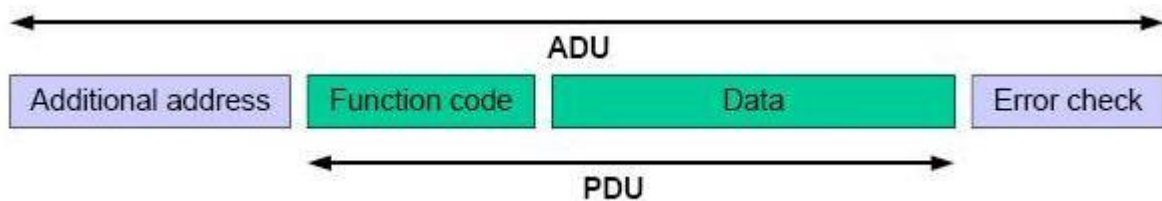
بین رشته کاراکترهای داخل پیام ممکن است فاصله زمانی تا یک ثانیه وجود داشته باشد. کاربرد با پیکر بندی Time Out میتواند مشخص کند که اگر زمان بیش از یک ثانیه باشد پیام خطا اعلام شود. برخی از شبکه های وسیع ممکن است به Time Out بین ۴ تا ۵ ثانیه نیاز داشته باشد.

هر بایت دیتا نیاز به ۲ کاراکتر برای کد کردن دیتا دارد بنابراین برای اطمینان از سازگاری بین مد های ASCII و RTU ماکزیمم سائز اختصاص یافته 2*252 کاراکتر است یعنی دو برابر سائز دیتای RTU. بدین ترتیب سائز کل فریم مطابق شکل بالا ۵۱۳ کاراکتر خواهد بود. کنترل خطا در این مد توسط LRC انجام میشود.

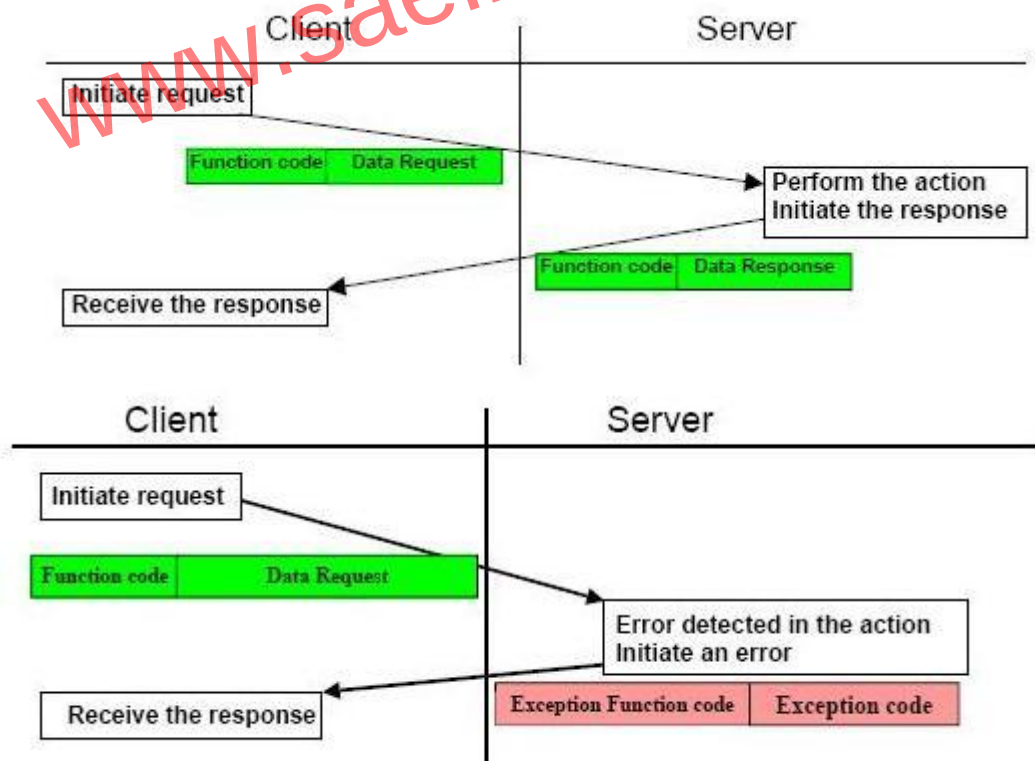
فانکشن کدهای Modbus

در ارتباط Modbus نوع عملی که باید انجام شود توسط فانکشن کدها مشخص میگردد. بعنوان مثال وقتی Master که بعضاً به آن Client هم گفته میشود بخواند از روی Slave که ممکن است Server هم نامیده شود I/O خاصی را بخواند این عمل را توسط کد خاصی که در فیلد فانکشن کد از فریم دیتا قرار می دهد درخواست میکند. این کد برای Slave نیز شناخته شده است از اینرو اطلاعات مورد نظر را با همان فانکشن کد به Master بر میگرداند.

قبل از اینکه به تشریح فانکشن کدها بپردازیم لازم است نحوه تبادل با Transaction دیتا روی شبکه Modbus تا حدی روشن شود تا جایگاه فانکشن کدها بهتر مشخص گردد. همانطور که قبلاً نیز اشاره شد در فریم دیتا علاوه بر فیلد دیتا فیلد های دیگری نیز وجود دارند. اگر فیلدهای شروع و پایان و فیلد آدرس را در نظر بگیریم با بسته PDU مخفف Protocol Data Unit و بسته ADU مخفف Application Data Unit مانند شکل زیر سرو کار داریم.



فیلد Function Code یک بایت است که می تواند بین ۱ تا ۲۵۵ دسیمال باشد. کدهای ۱ تا ۱۲۷ برای کار نرمال و کدهای ۱۲۸ تا ۲۵۵ 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 در خواست Master را انجام داده و در پاسخ خود به همان کد فانکشن اشاره میکند اصطلاحاً گفته میشود که 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 کد فانکشن کد Exception آمده برگشت داده میشود تا Master از بروز خطا مطلع شود. شکل های بعد این دو حالت را نمایش میدهد.



فانکشن کدها را میتوان به ۲ دسته کلی تقسیم کرد :

۱ - فانکشن های عمومی (Public) این فانکشن ها بصورت استاندارد از قبل تعریف شده هستند و برای مقاصد مشخص مانند خواندن و نوشتن وسیله بکار میرود .

۲ - فانکشن های خاص (User Defined) این فانکشن ها می توانند توسط کاربر تعریف شوند و نیازی به تایید موسسه Modbus.org ندارند ولی باید توجه داشت که کدهای رزرو شده را برای این فانکشنها نمیتوان استفاده کرد . کد فانکشنهای کاربر میتواند در محدوده ۶۵ تا ۷۲ یا ۱۰۰ تا ۱۱۰ باشد .

