

DeviceNet Protocol

الهه مرتضويان

فهرست

۳ DeviceNet چیست؟
۳ مدل OSI
۳ لایه فیزیکی
۴ لایه لینک داده
۴ اشتراک باس
۵ فرمت پیام
۶ لایه شبکه و انتقال
۷ لایه‌های بالاتر
۷ مجموعه ارتباط‌های Controller/Device
۸ نتیجه‌گیری
۸ مراجع

DeviceNet چیست؟

DeviceNet یک شبکه صنعتی مبتنی بر فناوری CAN¹ می باشد که برای ارتباط بین دستگاه های هوشمند در اتوماسیون صنعتی است. این شبکه در لایه های بالا از CIP² استفاده می کند و امکان تبادل داده بین کنترلرها، سنسورها و عملگرها را فراهم می کند. DeviceNet از معماری تولیدکننده-مصرف کننده استفاده می کند که باعث بهینه سازی ارتباط بین دستگاه های مختلف می شود. در این مدل دستگاه ها می توانند یا کلاینت باشند و یا سرور و هر سرور و کلاینت می تواند یک مصرف کننده باشد یا تولید کننده یا هر دو باشند. تولید کننده یک درخواست را تولید می کند و مصرف کننده پاسخ هایی که دریافت می کند را مصرف می کند.

از ویژگی های مهم این پروتکل می توان به این اشاره کرد که امکان پشتیبانی از حداکثر ۶۴ گره، امکان انتقال داده تا نرخ حداکثر ۵۰۰ کیلوبیت بر ثانیه و تحمل بالا برای نویز را دارد. این شبکه با فراهم کردن یک زیرساخت انعطاف پذیر و کم هزینه امکان کنترل و مدیریت آسان دستگاه ها را در محیط های صنعتی فراهم می کند. در ادامه این مستند با بخش های مختلف این پروتکل بیشتر آشنا خواهیم شد.

مدل OSI

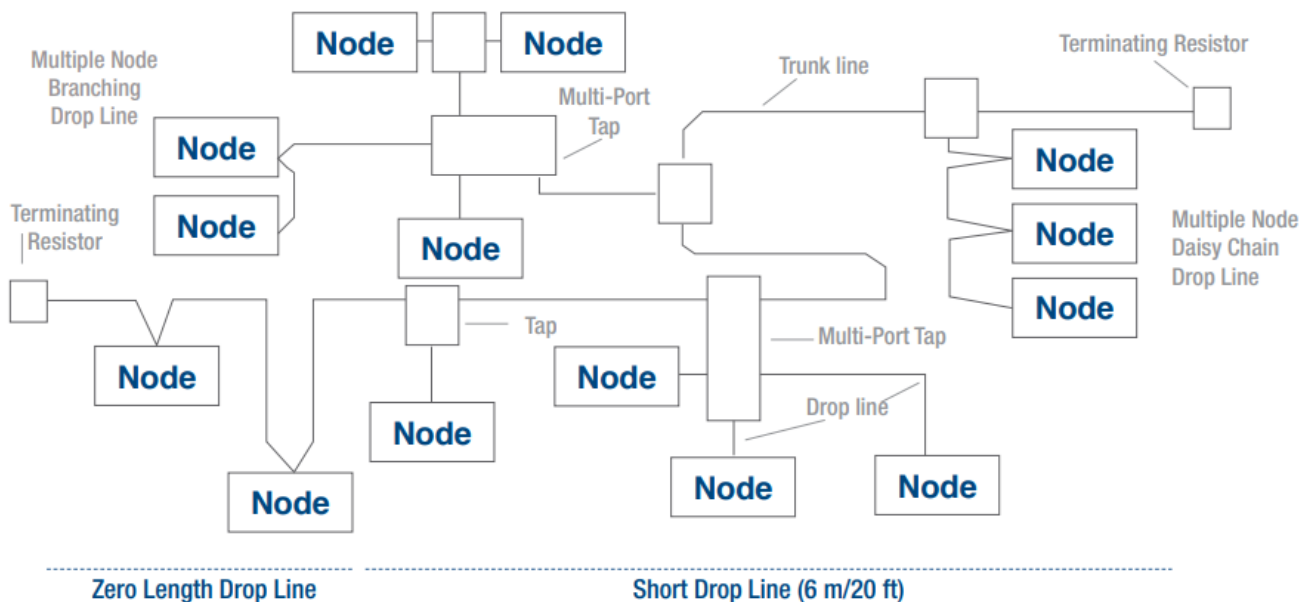
DeviceNet از مدل OSI پیروی می کند و برای لایه های مختلف رویه های مشخصی را رعایت می کند. در این بخش به بررسی عملکرد DeviceNet در لایه های مختلف می پردازیم.

لایه فیزیکی

DeviceNet در لایه فیزیکی از توپولوژی trunkline-dropline پیروی می کند که طبق آن کاربر می تواند برای سیگنال های سیگنال و توان از باس های twisted-pair جدا استفاده کند. این کار باعث می شود از پیچیدگی سیم کشی کم شود و هزینه پیاده سازی بهینه شود. یک مثال از پیاده سازی این توپولوژی را در شکل زیر مشاهده می کنید. در لایه فیزیکی می توان تا حداکثر ۶۴ گره را اضافه یا جدا کرد. همچنین برای اضافه کردن power tap محدودیتی نداریم و می توان در هر نقطه اضافه شوند. جریان trunkline با توجه به نوع کابل برابر ماکسیمم ۸ آمپر است.

¹ Controller Area Network

² Common Industrial Protocol



شکل ۱ توپولوژی در لایه فیزیکی

به طور کلی DeviceNet از سه نرخ داده ممکن پشتیبانی می‌کند و کاربر می‌تواند با توجه به نیاز خود کابل مورد نیاز برای این نرخ‌ها را انتخاب کند که حداکثر نرخ می‌تواند ۵۰۰ کیلوبیت بر ثانیه باشد. DeviceNet امکان استفاده از باس مشترک را برای دستگاه‌هایی که از یک منبع بیرونی تغذیه می‌شوند را نیز فراهم می‌کند که به صرفه‌جویی در فضا و کاهش پیچیدگی سیم کمک می‌کند.

لایه لینک داده^۳

برای این لایه، DeviceNet از پروتکل CAN استفاده می‌کند که به دلیل سربار^۴ کم خود در این لایه به انتقال و مدیریت پیام‌ها به صورت بهینه کمک می‌کند. برای پکیج کردن و انتقال پیام‌های CIP از کمترین مقدار پهنای باند استفاده می‌شود و همچنین برای پارس کردن پیام‌ها سعی می‌شود تا کمترین سربار را روی پردازنده داشته باشد.

اشتراک باس

از پروتکل CAN برای کنترل باس نیز استفاده می‌شود. این پروتکل دو حالت **dominant (logic 0)** و **recessive (logic 1)** را تعریف می‌کند. هر دستگاهی می‌تواند برای انتقال باس را در حالت **dominant** قرار دهد و باس تنها در صورتی در حالت **recessive** خواهد بود که هیچ دستگاهی برای انتقال در حالت **dominant** نباشد. از این تعریف برای **bus arbitration** استفاده می‌شود. هر گره یا دستگاه فقط وقتی یک پیام را انتقال می‌دهد که هیچ گره دیگری در

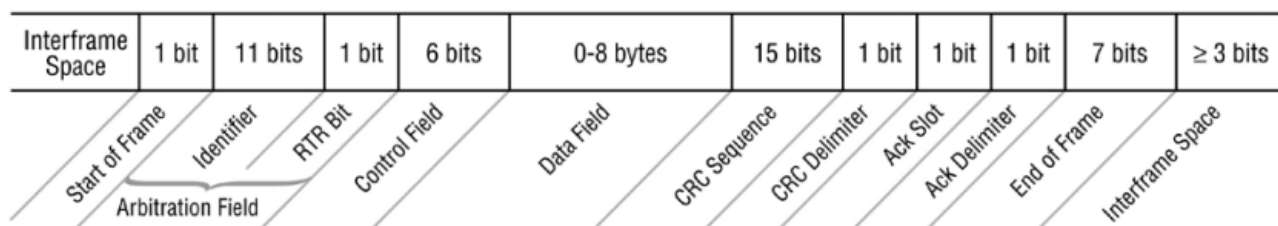
^۳ Data Link Layer

^۴ Overhead

حال انتقال نباشد. CAN از روش bitwise arbitration برای رفع تداخل‌های باس استفاده می‌کند که باعث می‌شود داده‌ای از دست نرود و پهنای باند حفظ شود. در بخش بعدی نحوه انجام این مکانیزم را شرح می‌دهیم.

فرمت پیام

از بین انواع مختلف فریم‌ها در پروتکل CAN، در DeviceNet تنها از data frame استفاده می‌شود که فرمت آن را در شکل زیر مشاهده می‌کنید.



شکل ۲ فرمت پیام در لایه لینک داده

- Start frame: وقتی یک بیت start frame فرستاده می‌شود همه گیرنده‌ها در شبکه به حالت dominant تغییر حالت می‌دهند.
- Identifier و RTR bit⁵: این دو فیلد با هم فیلد arbitration را تشکیل می‌دهند که از آن برای تعیین اولویت استفاده می‌شود. بیتی اولویت بالاتری دارد که کمترین مقدار را داشته باشد. Identifier به عنوان یک شناسه برای مشخص کردن هر دستگاه استفاده می‌شود. این ۱۱ بیت به ۴ گروه تقسیم می‌شوند. ۳ گروه اول شامل دو فیلد برای MAC ID و Message ID است و ترکیب این فیلدها شناسه را تشکیل می‌دهد. از این فیلد همچنین به منظور مشخص کردن شناسه‌های دستگاه تکراری نیز استفاده می‌شود. نحوه تخصیص این فیلدها را در شکل زیر مشاهده می‌کنید. در DeviceNet هر گره مسئول شناسه خود است و این شناسه توسط واحد دیگری اختصاص داده نمی‌شود. این کار باعث می‌شود اضافه کردن و حذف کردن گره در شبکه آسان‌تر شود.

⁵ Remote Transmission Request bit

IDENTIFIER BITS											HEX RANGE	IDENTITY USAGE
10	9	8	7	6	5	4	3	2	1	0		
0	Group 1 Message ID				Source MAC ID						000 – 3ff	Message Group 1
1	0	MAC ID						Group 2 Message ID			400 – 5ff	Message Group 2
1	1	Group 3 Message ID			Source MAC ID						600 – 7bf	Message Group 3
1	1	1	1	1	Group 4 Message ID (0-2f)						7c0 – 7ef	Message Group 4
1	1	1	1	1	1	1	X	X	X	X	7f0 – 7ff	Invalid CAN Identifiers
10	9	8	7	6	5	4	3	2	1	0		

شکل ۳ شناسه CAN

- Control Field: دو بیت فیکس دارد و ۴ بیت دیگر تعیین کننده اندازه داده انتقالی است.
- Data field: بیت‌های داده که از صفر تا ۸ بایت می‌تواند باشد. اگر مقدار داده از ۸ بایت بیشتر باشد از Fragmentation استفاده می‌شود.
- CRC⁶ field: برای تشخیص خطا در فریم‌ها استفاده می‌شود. همچنین CAN با انجام تلاش‌های دوباره به صورت اتومات می‌تواند یک عملکرد ایمن و قابل اطمینان را فراهم کند.
- ACK field: برای جلوگیری از گم شدن فریم‌ها استفاده می‌شود.
- End of frame: برای مشخص کردن پایان پیام.

لایه شبکه^۷ و انتقال^۸

DeviceNet یک شبکه مبتنی بر ارتباط^۹ است که یعنی باید برای انتقال داده ابتدا یک ارتباط برقرار شود. این برقراری ارتباط از دو روش امکان‌پذیر است.

⁶ Cyclic Redundancy Check

⁷ Network Layer

⁸ Transport Layer

⁹ Connection based

- روش ¹⁰UCMM: در این روش اگر یک ماژول بخواهد با یک ماژول دیگر ارتباط برقرار کند یک پیغام ¹¹Explicit را ارسال می‌کند که MAC ID ماژول دیگر در پیام داده شده است. ماژول دیگر شناسه خود را در فیلد پیام تشخیص می‌دهد و یک پیام صریح می‌فرستد و ارتباط برقرار می‌شود.
 - روش Group 2 Unconnected port: برای دستگاه‌هایی استفاده می‌شود که نیاز به ارتباط پیوسته ندارند و فقط یک بار انتقال پیام کافی می‌باشد.
- DeviceNet از دو نوع پیغام پشتیبانی می‌کند:
- Explicit: برای برقراری ارتباط اولیه استفاده می‌شود. بیشتر برای تراکنش‌های request/response استفاده می‌شود. این پیغام می‌تواند از طریق هر دو نوع روش ارتباط فرستاده شود.
 - Implicit: برای انتقال پیام‌های I/O استفاده می‌شود.

لایه‌های بالاتر

برای لایه‌های بالاتر DeviceNet از CIP استفاده می‌کند که یک پروتکل شی‌گراست. هر شی در CIP دارای صفات (داده)، سرویس (دستور) و رفتارهایی (واکنش) است. مدل تولیدکننده-مصرف‌کننده در CIP این امکان را می‌دهد که انتقال داده از یک فرستنده به چند گیرنده ممکن باشد و نیازی به انتقال داده به صورت تکراری نباشد. در مدل‌های تولیدکننده-مصرف‌کننده یک پیغام توسط شناسه ارتباطش شناخته می‌شود و نه آدرس مقصدش.

CIP برای هر دستگاه یک Device Profile تعریف می‌کند که شامل نوع دستگاه، کانفیگ آن و نیازمندی‌هایش است. این کار باعث می‌شود که استفاده‌کنندگان CIP یک واسطه یکسان برای دستگاه‌های مختلف داشته باشند و بتوان از یک شبکه CIP (مثل DeviceNet) به شبکه‌های CIP دیگر (مثل Ethernet) انتقال داده انجام داد و ارتباط برقرار کرد. برای این ارتباط بین شبکه‌ای با استفاده از CIP نیازی به لایه‌های Presentation یا اپلیکیشن نیست.

مجموعه ارتباط‌های Controller/Device

برای ساده کردن ارتباط‌ها و پکیج کردن پیام‌ها، DeviceNet از یک Predefined Controller/Device set استفاده می‌کند. بعضی از سنسورها و حسگرها که مختص DeviceNet ساخته شده‌اند این ویژگی را دارند که هنگام

¹⁰ Unconnected Message Manager

¹¹ Unconnected Explicit Request Message

روشن شدن^{۱۲} یک روند یا کارکرد از پیش تعیین شده‌ای را انجام دهند. مجموعه Controller/Device از پیش تعیین شده یک سری اشیا ارتباط را فراهم می‌کنند که در زمان روشن شدن دستگاه کانفیگ می‌شوند.

دستگاه‌ها می‌توانند سه نوع مختلف پیام را استفاده کنند. این انتخاب با توجه به نیازمندی‌های اپلیکیشن مشخص می‌شود. انواع پیام‌های استفاده شده عبارتند از:

- **Polled**: دستگاه به صورت دنباله‌ای داده را از سمت کنترلر دریافت می‌کند. ترتیب این دنباله توسط لیست اسکن کنترلر تعیین می‌شود. نرخ **Polling** کنترلر توسط تعداد گره‌ها در لیست اسکن، **Baud rate**، اندازه پیام‌ها و زمان‌بندی داخلی کنترلر مشخص می‌شود. این نوع پیام باعث می‌شود شاهد یک رفتار قطعی^{۱۳} در سیستم باشیم. در این نوع روش داده خروجی به صورت **unicast** و داده ورودی به صورت **multicast** است.
- **Cyclic**: دستگاه به صورت دوره‌ای پیام‌ها را تولید می‌کند و این کار در بازه‌های یکسان و مشخصی انجام می‌شود. این نوع پیام به کاربران این اجازه را می‌دهد تا در یک نرخ مناسب داده تولید کنند که بسته به نوع کاربرد می‌تواند باعث کاهش ترافیک روی سیم شود و استفاده بهینه‌ای از پهنای باند را ممکن کند.
- **Change-of-state**: دستگاه در زمانی که داده دچار تغییرات شود، پیام را تولید می‌کند. **DeviceNet** برای این نوع پیام یک محدودیت زمانی قرار می‌دهد تا اگر پیام‌ها در یک نرخ سریع تولید شدند از وقوع **flooding** جلوگیری کند.

نتیجه‌گیری

به طور کلی پروتکل **DeviceNet** یک پروتکل بسیار منعطف است که برای استفاده در فضاهای صنعتی مناسب است. این پروتکل به کاربران خود اجازه کنترل سطوح مختلف نرخ داده، کابل‌های استفاده شده و تعداد گره‌های متصل را می‌دهد. همچنین ویژگی‌های منحصر به فرد این پروتکل باعث می‌شود قابلیت تغییر داشته باشد و اضافه کردن دستگاه جدید به آن هزینه و چالش برانگیز نباشد که برای یک محیط صنعتی حیاتی است.

مراجع

Elprocus. n.d. *DeviceNet : Architecture, Message Format, Error Codes, Working & Its Applications*.
<https://www.elprocus.com/devicenet-architecture/>.

ODVA. n.d. https://www.odva.org/wp-content/uploads/2021/05/PUB00026R5_Tech-Series-DeviceNet.pdf.

¹² Power up

¹³ Deterministic

—. n.d. https://www.dia.uniroma3.it/autom/Reti_e_Sistemi_Automazione/PDF/DNET-Overview.pdf.

—. n.d. *DeviceNet ODVA*. <https://www.odva.org/technology-standards/key-technologies/devicenet/>.

Realpars. n.d. *What is DeviceNet?* <https://www.realpars.com/blog/devicenet>.