

«به نام خدا»
دانشکده مهندسی کامپیوتر
دانشگاه صنعتی شریف
زمستان ۱۴۰۴



بررسی جامع استانداردهای ارتباطی در سیستم‌های نهفته

هانیه سادات میرعمادی

دانشجوی کارشناسی مهندسی کامپیوتر دانشگاه صنعتی شریف

چکیده

این تحقیق، به صورت جامع، استانداردهای ارتباطی پرکاربرد در سیستم‌های نهفته را از منظر معماری پروتکل، لایه فیزیکی، زمان‌بندی انتقال، قابلیت اطمینان، و ملاحظات طراحی سخت‌افزار واسط بررسی و مقایسه می‌کند.

ابتدا رابط‌های کم‌هزینه و رایج برد-به-برد تحلیل می‌شوند: UART به عنوان ارتباط ناهمزمان مبتنی بر فریم‌بندی (بیت شروع/پایان و سرعت) برای دیباگ، کنسول و لینک‌های ساده معرفی می‌شود [1]. سپس SPI به عنوان گذرگاه همزمان تمام‌دوطرفه با خطوط کلاک/داده و «انتخاب تراشه» برای ارتباطات سریع کوتاه‌برد (حافظه‌ها، ADC/DAC و سنسورها) بررسی می‌گردد و اثر حالت‌های کلاک و توپولوژی چند-هدفی روی طراحی مطرح می‌شود [2].

در ادامه I²C به عنوان باس دو سیمه با آدرس‌دهی، داوری (arbitration) و حالت‌های سرعت مختلف (استاندارد تا High-speed و...)، همراه با محدودیت‌های خازن باس و نقش مقاومت‌های Pull-up تحلیل می‌شود [3].

برای کاربردهای خودرو و صنعتی، CAN به عنوان پروتکل مقاوم در برابر نویز با سیگنال‌دهی تفاضلی، زمان‌بندی بیت و سازوکارهای تشخیص/مدیریت خطا (در سطح استاندارد و اسناد پروتکل) مقایسه می‌شود [4].

همچنین LIN به عنوان گزینه کم‌هزینه‌تر برای زیرشبکه‌های بدنه خودرو (ساختار فریم، زمان‌بندی و

الزامات لایه فیزیکی) و جایگاه آن در معماری‌های ترکیبی CAN/LIN بررسی می‌گردد [5]. در بخش رابط‌های پرسرعت و تجهیزات جانبی، USB (به‌ویژه USB 2.0) از منظر لایه فیزیکی، نقش میزبان/دستگاه و الزامات سازگاری/پیچیدگی پیاده‌سازی مرور می‌شود [6]. در نهایت، چالش‌های طراحی مدار واسط-از جمله تطبیق سطح ولتاژ و مبدل‌های سطح، Signal Integrity (بازتاب، overshoot/undershoot و اثر مسیره‌های PCB)، و حفاظت ESD/Surge مخصوصاً برای پورت‌های بیرونی-بیان می‌شود تا معیارهای انتخاب رابط مناسب (سرعت، فاصله، نویز پذیری، توپولوژی و هزینه) به‌صورت کاربرد محور استخراج گردد.

۱- مقدمه

سیستم‌های نهفته به‌عنوان هسته اصلی محصولات هوشمند در حوزه‌های صنعتی، پزشکی، مخابراتی، خودرویی و اینترنت اشیاء، نیازمند تبادل داده‌ای مطمئن، کم‌هزینه و سازگار با محدودیت‌های توان و فضا هستند. بخش عمده‌ای از عملکرد این سیستم‌ها بر پایه ارتباط بین میکروکنترلرها، سنسورها، عملگرها و تجهیزات جانبی شکل می‌گیرد؛ ازاین‌رو انتخاب و طراحی صحیح استانداردهای ارتباطی، نقشی تعیین‌کننده در کارایی، پایداری و قابلیت اطمینان کل سامانه دارد. پروتکل‌هایی مانند UART، SPI، I²C، CAN، LIN و USB هر یک با معماری، لایه فیزیکی و سازوکارهای کنترلی متفاوت، برای پاسخ‌گویی به نیازهای خاصی توسعه یافته‌اند و شناخت دقیق ویژگی‌ها، محدودیت‌ها و الزامات طراحی آن‌ها، پیش‌نیاز توسعه سیستم‌های مدرن است.

با گسترش کاربردهای بلادرنگ، افزایش نرخ داده و الزامات ایمنی عملکردی در صنایع (به‌ویژه خودرو و صنعت)، صرفاً شناخت تئوریک پروتکل‌ها کافی نیست؛ بلکه تحلیل یکپارچه‌ای از ساختار پروتکل، روش‌های زمان‌بندی، تحمل‌پذیری در برابر نویز، توپولوژی شبکه و چالش‌های طراحی سخت‌افزار واسط-از جمله تطبیق سطح ولتاژ، یکپارچگی سیگنال (Signal Integrity)، ایزولاسیون و حفاظت ESD-ضروری است. در بسیاری از پروژه‌های عملی، انتخاب نامناسب رابط ارتباطی می‌تواند منجر به افزایش هزینه، کاهش قابلیت اطمینان یا پیچیدگی بیش از حد طراحی شود. بنابراین، پژوهشی جامع که این استانداردها را از منظر مهندسی سیستم و طراحی مدار مقایسه کند، خلأ مهمی را پوشش می‌دهد.

در ادبیات علمی و صنعتی، مطالعات متعددی به بررسی تک‌پروتکل‌ها پرداخته‌اند؛ برای نمونه، اسناد فنی شرکت‌هایی نظیر NXP، Texas Instruments و Microchip به تشریح معماری و الزامات

پیاده‌سازی رابط‌هایی مانند I²C و SPI پرداخته‌اند. در حوزه شبکه‌های صنعتی و خودرویی، پژوهش‌های گسترده‌ای دربارهٔ CAN و LIN با تمرکز بر تحمل خطا، زمان‌بندی قطعی (Deterministic Timing) و تحلیل خطا منتشر شده است. همچنین مقالات دانشگاهی متعددی به مقایسهٔ عملکرد گذرگاه‌های سریال از نظر سرعت، مصرف توان و قابلیت اطمینان پرداخته‌اند. با این حال، اغلب این منابع یا بر یک پروتکل خاص متمرکز بوده‌اند یا ملاحظات سخت‌افزاری و یکپارچگی سیگنال را به‌صورت محدود بررسی کرده‌اند.

هدف این تحقیق، ارائهٔ یک چارچوب تحلیلی جامع برای مقایسه و ارزیابی استانداردهای ارتباطی در سیستم‌های نهفته است؛ به‌گونه‌ای که علاوه بر بررسی ساختار و عملکرد پروتکل‌ها، معیارهای کاربردی برای انتخاب مناسب‌ترین رابط در سناریوهای مختلف (مصرف پایین، محیط نویزی، فاصله زیاد، نرخ داده بالا و غیره) ارائه شود. این پژوهش می‌کوشد با تلفیق دیدگاه نظری و عملی، راهنمایی نظام‌مند برای مهندسان طراح فراهم آورد تا تصمیم‌گیری آگاهانه‌تری در انتخاب و پیاده‌سازی رابط‌های ارتباطی داشته باشند.

۲- بدنه‌ی تحقیق

در این بخش، استانداردهای ارتباطی پرکاربرد در سیستم‌های نهفته از منظر ساختار پروتکل، لایه فیزیکی، مکانیزم انتقال داده، مدیریت خطا، توپولوژی شبکه، سرعت، نویز پذیری و چالش‌های طراحی سخت‌افزار بررسی و تحلیل می‌شوند. تمرکز اصلی بر درک عملی تفاوت‌ها و شباهت‌های این رابط‌هاست تا بتوان معیارهای فنی انتخاب هر پروتکل را در کاربردهای مختلف استخراج کرد.

همچنین در بخش دوم، هر استاندارد علاوه بر معرفی مفهومی، از دید مهندسی سیستم نیز ارزیابی می‌شود؛ به‌ویژه تأثیر آن بر طراحی PCB، یکپارچگی سیگنال (Signal Integrity)، تطبیق سطح ولتاژ، نیاز به ایزولاسیون و حفاظت در برابر ESD.

۲-۱- استانداردهای ارتباطی پرکاربرد

۲-۱-۱- استاندارد UART (Universal Asynchronous Receiver/Transmitter)

UART ساده‌ترین و پایه‌ای‌ترین رابط سریال در سیستم‌های نهفته محسوب می‌شود و به دلیل ساختار

کم‌هزینه و پیاده‌سازی آسان، در طیف گسترده‌ای از سامانه‌های الکترونیکی به کار می‌رود. این پروتکل برای ارتباطات کوتاه‌برد و غیرشبکه‌ای گزینه‌ای مناسب است، زیرا به سخت‌افزار پیچیده یا لایه‌های کنترلی پیشرفته نیاز ندارد. با این حال، در شبکه‌های چندگانه‌ای که نیازمند آدرس‌دهی و داوری بین گره‌ها هستند، در محیط‌های صنعتی با نویز الکترومغناطیسی بالا، در کاربردهای بلادرنگ با الزام مدیریت خطای پیشرفته و همچنین در فواصل طولانی بدون استفاده از لایه فیزیکی تقویت‌شده، انتخاب مناسبی به شمار نمی‌رود. در مقابل، زمانی که سادگی نسبت به سرعت اولویت داشته باشد، هزینه پایین اهمیت داشته باشد، ارتباط نقطه‌به‌نقطه کافی باشد یا نیاز به دیباگ سریع وجود داشته باشد، UART انتخابی منطقی است. این رابط در هرم استانداردهای ارتباطی، پایه‌ای‌ترین گزینه محسوب می‌شود و در کنار پروتکل‌های پیشرفته‌تر مانند CAN و USB نقش مکمل دارد.

در ادامه، جزئیات لازم ذکر شده است:

❖ ساختار پروتکل:

UART یک رابط سریال ناهمزمان است که انتقال داده را بدون استفاده از سیگنال کلاک مشترک انجام می‌دهد. هر فریم داده با یک بیت شروع در سطح منطقی صفر آغاز می‌شود و پس از آن بیت‌های داده که معمولاً بین پنج تا نه بیت هستند ارسال می‌گردند که رایج‌ترین حالت آن هشت بیت است. در صورت نیاز، یک بیت توازن برای تشخیص خطای ساده افزوده می‌شود و در پایان یک یا دو بیت پایان در سطح منطقی یک قرار می‌گیرد. قالب متداول در بسیاری از کاربردها 8N1 است که شامل هشت بیت داده، بدون بیت توازن و یک بیت پایان است. در این پروتکل هیچ سازوکار آدرس‌دهی داخلی وجود ندارد و ارتباط صرفاً به صورت نقطه‌به‌نقطه انجام می‌شود؛ بنابراین هرگونه مدیریت سطح بالاتر مانند بسته‌بندی داده یا کنترل صحت انتقال باید در نرم‌افزار پیاده‌سازی شود.

❖ نحوه عملکرد لایه فیزیکی:

UART تنها قالب داده را تعریف می‌کند و سطح ولتاژ مشخصی را الزام نمی‌کند. در عمل، رایج‌ترین پیاده‌سازی آن در سطح TTL یا CMOS با ولتاژهای صفر تا 3.3 یا 5 ولت است که برای ارتباط مستقیم بین میکروکنترلرها استفاده می‌شود. در ارتباط با رایانه‌های شخصی از استاندارد RS-232 استفاده می‌شود که از سطوح ولتاژ مثبت و منفی در بازه تقریبی سه تا پانزده ولت بهره می‌برد و به همین دلیل به مبدل سطح مانند MAX232 نیاز دارد. در کاربردهای صنعتی نیز می‌توان UART را بر بستر RS-485 پیاده‌سازی کرد که از سیگنال‌دهی تفاضلی استفاده می‌کند و امکان انتقال در فواصل طولانی‌تر و محیط‌های نویزی را فراهم می‌سازد. در حالت TTL که انتقال به صورت تک‌پایانه انجام می‌شود، حساسیت به نویز نسبت به ارتباطات تفاضلی بیشتر است.

❖ روش‌های زمان‌بندی و انتقال داده:

از آنجا که UART ناهمزمان است، همگام‌سازی تنها از طریق تشخیص بیت شروع انجام می‌شود. گیرنده با شناسایی لبه نزولی بیت شروع، فرآیند نمونه‌برداری را آغاز می‌کند. برای افزایش دقت، معمولاً از روش نمونه‌برداری با ضریب بالاتر از نرخ بود، مانند شانزده برابر، استفاده می‌شود تا موقعیت دقیق هر بیت مشخص گردد. اختلاف مجاز بین فرکانس کلاک فرستنده و گیرنده معمولاً کمتر از حدود دو درصد است، بنابراین دقت اسیلاتور داخلی اهمیت زیادی دارد. این پروتکل می‌تواند به صورت نیمه‌دوطرفه با یک خط مشترک یا تمام‌دوطرفه با خطوط مجزای ارسال و دریافت عمل کند.

❖ سرعت (Baud Rate):

سرعت انتقال که با نرخ بود بیان می‌شود، در کاربردهای عمومی معمولاً بین 300 بیت بر ثانیه تا 115200 بیت بر ثانیه قرار دارد، اگرچه در میکروکنترلرهای مدرن می‌تواند به چند مگابیت بر ثانیه نیز برسد. سرعت عملی انتقال به عواملی مانند طول کابل، کیفیت سیگنال، دقت کلاک و سطح ولتاژ وابسته است و با افزایش فاصله، پایداری ارتباط کاهش می‌یابد.

❖ توپولوژی شبکه:

UART ذاتاً یک ارتباط نقطه‌به‌نقطه میان یک فرستنده و یک گیرنده است و هیچ پشتیبانی داخلی برای ساختار چندگره‌ای ارائه نمی‌دهد. برای ایجاد شبکه‌ای با چند گره، باید از تجهیزات جانبی مانند مالتی‌پلکسر استفاده شود یا این پروتکل بر بستر RS-485 همراه با یک لایه نرم‌افزاری برای مدیریت گره‌ها پیاده‌سازی گردد.

❖ مزایا:

سادگی سخت‌افزار، نیاز به سیم‌کشی محدود، پشتیبانی گسترده در تقریباً تمام میکروکنترلرها و مناسب بودن برای کاربردهایی مانند دیباگ و بوت‌لودر از مهم‌ترین مزایای UART به شمار می‌روند.

❖ محدودیت‌ها:

نبود مکانیزم پیشرفته تشخیص و تصحیح خطا، حساسیت به اختلاف نرخ کلاک، عدم مقیاس‌پذیری شبکه، نویز پذیری بالا در سطح TTL و کاهش بازده انتقال به دلیل وجود بیت‌های شروع و پایان از جمله محدودیت‌های این پروتکل محسوب می‌شوند.

❖ میزان نویز پذیری:

در حالت TTL میزان حساسیت نسبتاً بالا است زیرا انتقال به صورت تک‌پایانه انجام می‌شود. در RS-232 عملکرد بهتر می‌شود، اما بیشترین مقاومت در برابر نویز زمانی حاصل می‌شود که UART بر بستر RS-485 با سیگنال‌دهی تفاضلی پیاده‌سازی شود. به دلیل نبود سازوکار تصحیح خطای پیشرفته، در محیط‌های صنعتی نویزی معمولاً همراه با لایه‌های حفاظتی و تدابیر تکمیلی استفاده می‌شود.

❖ موارد کاربرد:

UART در کنسول‌های دیباگ میکروکنترلر، ارتباط بین MCU و رایانه شخصی، ماژول‌های GPS، ماژول‌های GSM، ماژول‌های Bluetooth و همچنین در فرآیندهای Bootloader و به‌روزرسانی Firmware کاربرد گسترده‌ای دارد و همچنان یکی از بنیادی‌ترین رابط‌های ارتباطی در سامانه‌های نهفته محسوب می‌شود [7].

۲-۱-۲- استاندارد (Serial Peripheral Interface) SPI

SPI یک رابط پرسرعت و کم‌تأخیر برای ارتباطات کوتاه‌برد درون‌سیستمی محسوب می‌شود. سادگی آن موجب کاهش سربار پردازشی و افزایش کارایی انتقال داده می‌شود، اما فقدان قابلیت‌های پیشرفته شبکه‌ای و مدیریت خطا باعث می‌شود در کاربردهای صنعتی گسترده یا محیط‌های بسیار نویزی گزینه ایده‌آلی نباشد. انتخاب SPI زمانی منطقی است که سرعت بالا، سادگی و ارتباط نقطه‌به‌نقطه یا چند اسلیو محدود مدنظر باشد و فاصله انتقال کوتاه باشد. در مجموع، SPI مکملی برای پروتکل‌های پیچیده‌تر شبکه‌ای مانند CAN و USB محسوب می‌شود و جایگاه آن عمدتاً در سطح ارتباطات داخلی سیستم نهفته است.

❖ ساختار پروتکل:

SPI یک پروتکل ارتباط سریال همزمان است که به‌صورت Full-Duplex عمل می‌کند و برای تبادل داده بین یک Master و یک یا چند Slave طراحی شده است. ساختار پایه آن شامل چهار خط اصلی MOSI برای ارسال داده از مستر به اسلیو، MISO برای ارسال داده از اسلیو به مستر، SCLK برای کلاک تولیدشده توسط مستر و CS یا SS برای انتخاب دستگاه مقصد است. برخلاف I²C، این پروتکل دارای مکانیزم آدرس‌دهی داخلی نیست و انتخاب هر اسلیو از طریق فعال‌سازی خط Chip Select مجزا انجام می‌شود. قالب داده در SPI انعطاف‌پذیر است و معمولاً شامل فریم‌های 8 یا 16 بیتی می‌شود، اما استاندارد طول فریم ثابتی را تحمیل نمی‌کند و این موضوع به طراحی سخت‌افزار و تنظیمات کنترلر وابسته است.

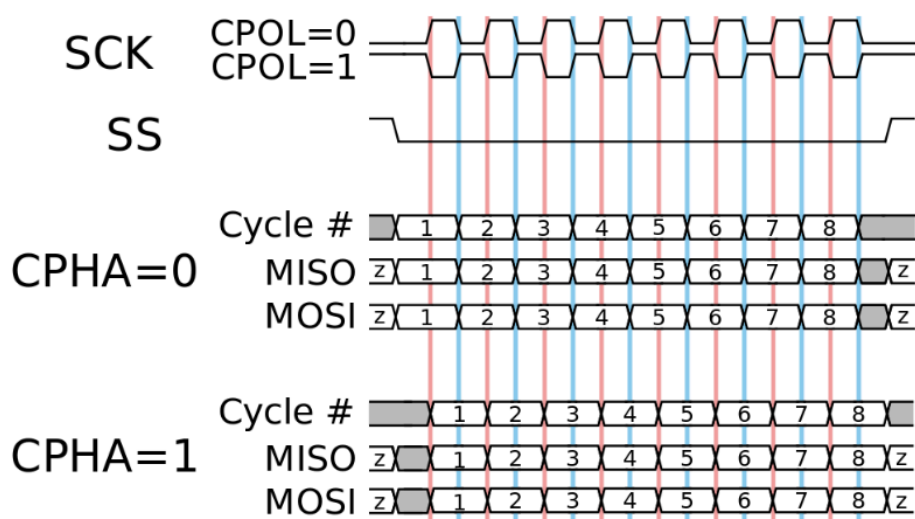
❖ نحوه عملکرد لایه فیزیکی:

در لایه فیزیکی، SPI معمولاً بر پایه سطوح منطقی TTL یا CMOS با ولتاژهای 3.3 یا 5 ولت پیاده‌سازی می‌شود. انتقال داده به‌صورت تک‌پایانه (Single-Ended) انجام می‌شود و کلاک توسط Master تولید می‌گردد. سیگنال‌ها به‌طور مستقیم بین دستگاه‌ها تبادل می‌شوند و هیچ درایور تفاضلی استاندارد در

SPI پیش‌بینی نشده است، بنابراین این پروتکل برای فواصل کوتاه روی برد یا بین بردهای نزدیک مناسب‌تر است. در سرعت‌های بالا، کنترل امپدانس مسیره‌های PCB، کاهش اعوجاج لبه سیگنال و جلوگیری از بازتاب اهمیت ویژه‌ای پیدا می‌کند. طول مسیره‌های MOSI و MISO و تطابق آن‌ها با سیگنال کلاک برای جلوگیری از خطای نمونه‌برداری بسیار مهم است.

❖ روش‌های زمان‌بندی و انتقال داده:

SPI یک پروتکل کاملاً سنکرون است و انتقال داده با هر پالس کلاک انجام می‌شود. داده‌ها معمولاً در لبه بالا رونده یا پایین رونده کلاک نمونه‌برداری می‌شوند که این موضوع توسط دو پارامتر CPOL و CPHA که در شکل زیر نمایش داده شده است، تعیین می‌شود. ترکیب این دو پارامتر چهار مد عملیاتی ایجاد می‌کند که به آن‌ها SPI Mode 0 تا Mode 3 گفته می‌شود. از آنجا که انتقال به صورت Full-Duplex انجام می‌شود، با هر بیت ارسال‌شده از سمت Master، همزمان یک بیت از سمت Slave دریافت می‌شود. نرخ انتقال داده مستقیماً تابع فرکانس کلاک است و می‌تواند از چند صد کیلوهرتز تا ده‌ها مگاهرتز افزایش یابد، بسته به توانایی سخت‌افزار و شرایط مسیر انتقال.



شکل ۱- مفهوم CPOL و CPHA در SPI

❖ مزایا و محدودیت‌ها:

SPI به دلیل ساختار ساده و عدم وجود سربار پروتکلی پیچیده، سرعت بسیار بالایی را با تأخیر کم فراهم می‌کند. این ویژگی آن را برای ارتباط با حافظه‌های Flash، مبدل‌های آنالوگ به دیجیتال، نمایشگرها و سنسورهای پرسرعت مناسب می‌سازد. در مقابل، نبود مکانیزم آدرس‌دهی داخلی باعث

افزایش تعداد خطوط سخت‌افزاری در صورت استفاده از چند Slave می‌شود. همچنین فاقد سیستم مدیریت خطا، داوری یا تصحیح خطای داخلی است و کنترل صحت داده باید در لایه نرم‌افزار انجام گیرد. این پروتکل ذاتاً مقیاس‌پذیری شبکه‌ای ندارد و برای سیستم‌های چندگره‌ای گسترده مناسب نیست.

❖ میزان نویز پذیری:

از آنجا که SPI مبتنی بر سیگنال تک‌پایانه است، نسبت به نویز الکترومغناطیسی حساس‌تر از پروتکل‌های تفاضلی مانند CAN یا RS-485 است. در سرعت‌های بالا، نویز، Crosstalk و بازتاب سیگنال می‌توانند موجب خطای بیت شوند. استفاده از مسیرهای کوتاه، لایه زمین مناسب و کنترل دقیق امپدانس در PCB برای حفظ یکپارچگی سیگنال ضروری است. در کاربردهای صنعتی با فاصله زیاد، معمولاً SPI توصیه نمی‌شود مگر با تمهیدات خاص تقویت سیگنال.

❖ سرعت:

سرعت SPI بسته به طراحی سخت‌افزار می‌تواند از چند صد کیلوبیت بر ثانیه تا بیش از 50 مگابیت بر ثانیه متغیر باشد. در بسیاری از میکروکنترلرهای مدرن، فرکانس کلاک SPI می‌تواند به کسری از فرکانس سیستم برسد. محدودیت سرعت معمولاً ناشی از کیفیت مسیر انتقال، ظرفیت خازنی خطوط و محدودیت‌های سخت‌افزاری Slave است.

❖ توپولوژی:

توپولوژی کلاسیک SPI به صورت Master-Slave است. یک Master می‌تواند چند Slave را کنترل کند، اما هر Slave نیازمند یک خط Chip Select مجزا است. در برخی طراحی‌ها از توپولوژی Daisy Chain استفاده می‌شود که در آن خروجی داده یک Slave به ورودی Slave بعدی متصل می‌شود. با این حال، این روش تأخیر تجمعی ایجاد می‌کند و پیچیدگی زمان‌بندی را افزایش می‌دهد.

❖ موارد کاربرد:

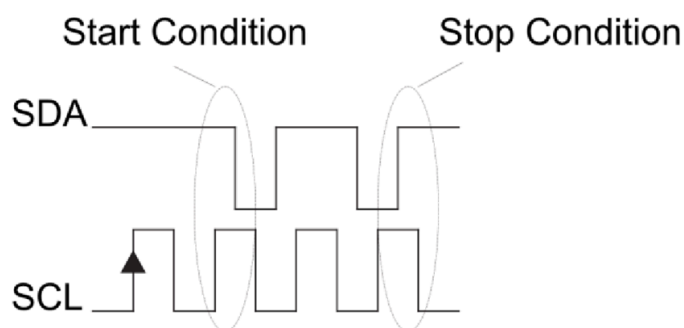
SPI در سیستم‌های نهفته برای ارتباط با حافظه‌های EEPROM و Flash، کارت‌های SD، نمایشگرهای TFT و OLED، سنسورهای با نرخ نمونه‌برداری بالا، مبدل‌های ADC و DAC و همچنین برخی ماژول‌های مخابراتی به کار می‌رود. به دلیل سرعت بالا و ساختار ساده، این پروتکل یکی از رایج‌ترین رابط‌های داخلی روی برد در تجهیزات الکترونیکی مدرن است [8].

۳-۱-۲- بررسی جامع استاندارد I²C (Inter-Integrated Circuit)

I²C یکی از پرکاربردترین گذرگاه‌های ارتباطی در سیستم‌های نهفته است که با هدف کاهش تعداد خطوط ارتباطی و ساده‌سازی اتصال چندین مدار مجتمع روی یک برد طراحی شده است. این پروتکل به دلیل استفاده از تنها دو خط سیگنال و قابلیت پشتیبانی از چندین گره، جایگاه ویژه‌ای در ارتباط با سنسورها، مبدل‌ها و تراشه‌های جانبی کم‌سرعت دارد. I²C تعادلی میان سادگی سخت‌افزار و قابلیت شبکه‌ای برقرار می‌کند؛ به‌گونه‌ای که نسبت به UART ساختارمندتر و نسبت به SPI کم‌سیم‌تر است، اما در عین حال از نظر سرعت و مقاومت در برابر نویز محدودیت‌هایی دارد. این استاندارد بیشتر برای ارتباطات کوتاه‌برد درون‌بردی مناسب است و در کاربردهای صنعتی با فاصله زیاد یا نرخ داده بالا معمولاً گزینه اول محسوب نمی‌شود.

❖ ساختار پروتکل:

I²C یک پروتکل سریال همزمان است که از دو خط اصلی SDA برای داده و SCL برای کلاک استفاده می‌کند (شکل ۲). هر دو خط به صورت Open-Drain یا Open-Collector پیاده‌سازی می‌شوند و از مقاومت‌های Pull-up برای بازگرداندن خط به سطح منطقی یک بهره می‌برند. ارتباط با یک شرط شروع (Start Condition) آغاز می‌شود که در آن خط داده در حالی که کلاک در سطح بالا قرار دارد از یک به صفر تغییر می‌کند. پس از آن، آدرس دستگاه مقصد که معمولاً هفت یا ده بیت است ارسال می‌شود و سپس بیت تعیین‌کننده خواندن یا نوشتن قرار می‌گیرد. هر بایت داده با یک بیت تأیید (ACK) دنبال می‌شود و در پایان انتقال، شرط توقف (Stop Condition) صادر می‌گردد. وجود مکانیزم داوری (Arbitration) امکان استفاده از چند Master را فراهم می‌کند، به‌گونه‌ای که در صورت همزمانی ارسال، گره‌ای که سطح خط را پایین‌تر نگه دارد اولویت خواهد داشت.



شکل ۲- SDA و SCL در استاندارد I²C

❖ نحوه عملکرد لایه فیزیکی:

در لایه فیزیکی، I^2C بر پایه سیگنال‌های تک‌پایانه با سطح ولتاژ منطقی 3.3 یا 5 ولت عمل می‌کند. ماهیت Open-Drain خطوط باعث می‌شود هر دستگاه تنها بتواند خط را به سطح صفر بکشد و بازگشت به سطح یک از طریق مقاومت Pull-up انجام شود. این ویژگی امکان اشتراک باس بین چندین دستگاه را بدون ایجاد تداخل مخرب فراهم می‌کند. با این حال، وجود خازن خط و مقدار مقاومت Pull-up مستقیماً بر شیب لبه سیگنال و زمان بالا رفتن آن تأثیر می‌گذارد. در نتیجه، طول باس و تعداد دستگاه‌های متصل محدود می‌شود، زیرا افزایش ظرفیت خازنی می‌تواند موجب کند شدن لبه‌ها و ایجاد خطا در نمونه‌برداری شود.

❖ روش‌های زمان‌بندی و انتقال داده:

از آنجا که I^2C همزمان است، کلاک توسط Master تولید می‌شود و تمام انتقال‌ها در هماهنگی با این سیگنال انجام می‌گیرد. داده‌ها در لبه مشخصی از کلاک پایدار بوده و در لبه دیگر نمونه‌برداری می‌شوند. یکی از ویژگی‌های مهم این پروتکل قابلیت Clock Stretching است که به دستگاه Slave اجازه می‌دهد در صورت نیاز به زمان بیشتر برای پردازش، خط کلاک را در سطح پایین نگه دارد و انتقال را موقتاً متوقف کند. این قابلیت انعطاف‌پذیری بالایی ایجاد می‌کند، اما در برخی طراحی‌ها ممکن است پیچیدگی زمان‌بندی را افزایش دهد. سرعت انتقال بسته به مد عملیاتی می‌تواند در حالت استاندارد 100 کیلوبیت بر ثانیه، در حالت سریع 400 کیلوبیت بر ثانیه، در Fast Mode Plus حدود 1 مگابیت بر ثانیه و در حالت High-Speed تا 3.4 مگابیت بر ثانیه باشد.

❖ سرعت:

سرعت I^2C در مقایسه با SPI پایین‌تر است، اما برای بسیاری از کاربردهای سنجش و کنترل کافی است. محدودیت سرعت عمدتاً ناشی از ویژگی‌های فیزیکی باس، به‌ویژه زمان بالا رفتن سیگنال و ظرفیت خازنی مسیر است. در طراحی‌های عملی، انتخاب مقدار مناسب مقاومت Pull-up و کوتاه نگه داشتن مسیرها برای دستیابی به سرعت‌های بالاتر ضروری است.

❖ توپولوژی شبکه:

I^2C از توپولوژی باس مشترک استفاده می‌کند که در آن تمامی دستگاه‌ها به دو خط مشترک متصل می‌شوند. این ساختار امکان اتصال چندین Slave به یک Master یا حتی چند Master را فراهم می‌کند. وجود آدرس‌دهی داخلی باعث می‌شود هر دستگاه تنها به پیام‌های مربوط به خود پاسخ دهد. با این حال، افزایش تعداد گره‌ها موجب افزایش ظرفیت خازنی باس شده و می‌تواند عملکرد را تحت تأثیر قرار دهد.

❖ مزایا:

کاهش تعداد سیم‌ها، امکان اتصال چندین دستگاه روی یک باس مشترک، پشتیبانی از آدرس‌دهی

داخلی و مصرف توان پایین از مهم‌ترین مزایای I²C به شمار می‌روند. همچنین تقریباً تمامی میکروکنترلرهای مدرن از این پروتکل پشتیبانی سخت‌افزاری دارند که پیاده‌سازی آن را ساده می‌کند.

❖ محدودیت‌ها:

سرعت نسبتاً پایین در مقایسه با SPI، حساسیت به ظرفیت خازنی باس، وابستگی به مقدار مقاومت Pull-up و حساسیت به نویز در مسیرهای طولانی از جمله محدودیت‌های I²C هستند. علاوه بر این، در صورت طراحی نادرست، پدیده‌هایی مانند گیر کردن باس در سطح پایین ممکن است رخ دهد که نیازمند مدیریت نرم‌افزاری است.

❖ میزان نویز پذیری:

به دلیل استفاده از سیگنال تک‌پایانه و مقاومت‌های Pull-up نسبتاً بزرگ، I²C در برابر نویز الکترومغناطیسی حساس‌تر از پروتکل‌های تفاضلی مانند CAN است. در محیط‌های صنعتی معمولاً طول باس کوتاه نگه داشته می‌شود یا از بافرها و ایزولاتورهای مخصوص I²C استفاده می‌شود تا پایداری ارتباط افزایش یابد.

❖ موارد کاربرد:

I²C به طور گسترده برای ارتباط با سنسورهای دما، فشار و شتاب، مبدل‌های آنالوگ به دیجیتال کم‌سرعت، حافظه‌های EEPROM، ساعت‌های واقعی (RTC)، نمایشگرهای کاراکتری و تراشه‌های مدیریت توان به کار می‌رود. این گستره کاربرد نشان می‌دهد که I²C به عنوان یک باس کنترلی و داده‌ای کم‌سرعت، نقش کلیدی در معماری داخلی سیستم‌های نهفته ایفا می‌کند [9].

۴-۱-۲- استاندارد CAN (Controller Area Network)

CAN یک پروتکل ارتباطی سریال چندگره‌ای و پیام‌محور است که در ابتدا برای صنعت خودرو توسعه یافت، اما به مرور در سامانه‌های صنعتی، پزشکی، ریلی و اتوماسیون نیز گسترش پیدا کرد. ویژگی شاخص CAN، مقاومت بالا در برابر نویز، قابلیت اطمینان بالا و وجود سازوکارهای پیشرفته تشخیص و مدیریت خطاست. برخلاف پروتکل‌هایی مانند UART یا SPI که اغلب برای ارتباط نقطه‌به‌نقطه به کار می‌روند، CAN به صورت ذاتی برای شبکه‌های چندگره‌ای طراحی شده و امکان ارتباط ده‌ها گره روی یک باس مشترک را فراهم می‌کند. این پروتکل در کاربردهای بلادرنگ که تأخیر قابل پیش‌بینی و قابلیت اطمینان اهمیت دارد، جایگاه ویژه‌ای دارد.

❖ ساختار پروتکل:

CAN یک پروتکل پیام‌محور است، به این معنا که به جای آدرس‌دهی به گره‌ها، پیام‌ها دارای شناسه (Identifier) هستند و هر گره بر اساس شناسه تصمیم می‌گیرد پیام را پردازش کند یا نادیده بگیرد.

فریم استاندارد CAN شامل بخش‌های Start of Frame، میدان داوری شامل شناسه پیام، بیت‌های کنترلی، میدان داده که در نسخه کلاسیک تا 8 بایت و در CAN FD تا 64 بایت است، بخش CRC برای تشخیص خطا، بیت تأیید (ACK) و End of Frame می‌شود. دو قالب شناسه وجود دارد که شامل قالب استاندارد با 11 بیت شناسه و قالب توسعه‌یافته با 29 بیت شناسه است. اولویت پیام‌ها بر اساس مقدار شناسه تعیین می‌شود و شناسه کوچکتر اولویت بالاتری دارد.

❖ نحوه عملکرد لایه فیزیکی:

در لایه فیزیکی، CAN از سیگنال‌دهی تفاضلی بین دو خط CAN_H و CAN_L استفاده می‌کند. در حالت غالب (Dominant)، اختلاف ولتاژ مشخصی بین این دو خط ایجاد می‌شود و در حالت مغلوب (Recessive)، هر دو خط تقریباً در سطح یکسان قرار دارند. این روش تفاضلی باعث افزایش مقاومت در برابر نویز الکترومغناطیسی و تداخل می‌شود. توپولوژی باس خطی است و در دو انتهای باس مقاومت‌های 120 اهم برای تطبیق امپدانس قرار می‌گیرند تا از بازتاب سیگنال جلوگیری شود. طول باس به نرخ داده وابسته است و با افزایش سرعت، حداکثر طول مجاز کاهش می‌یابد.

❖ روش‌های زمان‌بندی و انتقال داده:

CAN از مکانیزم داوری غیرمخرب (Non-Destructive Arbitration) استفاده می‌کند. در صورتی که چند گره به‌طور همزمان شروع به ارسال کنند، هر گره بیت ارسال‌شده را با سطح واقعی باس مقایسه می‌کند و در صورت مشاهده بیت غالب در حالی که بیت مغلوب ارسال کرده است، از رقابت کنار می‌رود. این فرآیند تضمین می‌کند که پیام با اولویت بالاتر بدون برخورد ارسال شود. زمان‌بندی بیت در CAN به بخش‌هایی مانند Sync Segment، Propagation Segment و Phase Segments تقسیم می‌شود که امکان تنظیم دقیق برای جبران تأخیر انتشار و نوسانات کلاک را فراهم می‌کند. این ساختار زمانی موجب افزایش پایداری شبکه در شرایط مختلف می‌شود.

❖ سرعت:

در نسخه کلاسیک CAN، سرعت انتقال می‌تواند تا 1 مگابیت بر ثانیه برسد، در حالی که در نسخه CAN FD نرخ داده در بخش داده می‌تواند به چند مگابیت بر ثانیه افزایش یابد. با این حال، سرعت بالاتر معمولاً به معنای کاهش طول مجاز باس است. برای مثال، در سرعت‌های پایین‌تر مانند 125 کیلوبیت بر ثانیه، می‌توان به طول‌های چند صد متر دست یافت، اما در سرعت 1 مگابیت بر ثانیه طول باس به چند ده متر محدود می‌شود.

❖ توپولوژی شبکه:

CAN از توپولوژی باس خطی استفاده می‌کند که در آن تمامی گره‌ها به یک زوج سیم مشترک متصل هستند. انشعابات باید کوتاه نگه داشته شوند تا بازتاب و اعوجاج سیگنال کاهش یابد. این ساختار

امکان اتصال تعداد زیادی گره را فراهم می‌کند و حذف یا اضافه شدن یک گره معمولاً تأثیر اساسی بر عملکرد سایر گره‌ها ندارد، مشروط بر رعایت اصول طراحی فیزیکی.

❖ مزایا:

مقاومت بسیار بالا در برابر نویز، وجود سازوکارهای پیشرفته تشخیص خطا شامل CRC، مانیتورینگ بیت، بررسی قالب فریم و تأیید دریافت، داوری غیرمخرب، قابلیت حذف خودکار گره‌های معیوب و پشتیبانی از شبکه‌های چندگره‌ای از مهم‌ترین مزایای CAN محسوب می‌شوند. این ویژگی‌ها آن را برای سامانه‌های ایمنی حساس بسیار مناسب کرده است.

❖ محدودیت‌ها:

نرخ داده محدود در نسخه کلاسیک، پیچیدگی بیشتر نسبت به پروتکل‌های ساده مانند UART و SPI، نیاز به طراحی دقیق باس و تطبیق امپدانس و همچنین سربار پروتکلی نسبتاً بالا از جمله محدودیت‌های CAN هستند. علاوه بر این، طول داده محدود در فریم کلاسیک ممکن است در برخی کاربردهای داده‌محور چالش ایجاد کند.

❖ میزان نویز پذیری:

به دلیل استفاده از سیگنال‌دهی تفاضلی و آستانه‌های ولتاژ تعریف‌شده، CAN در برابر نویز الکترومغناطیسی بسیار مقاوم است. این ویژگی به‌ویژه در محیط‌های خودرویی و صنعتی که تداخل الکتریکی شدید وجود دارد اهمیت دارد. همچنین سازوکارهای تشخیص و مدیریت خطا باعث می‌شوند در صورت بروز خطا، پیام مجدداً ارسال شود و گره‌های معیوب به حالت غیرفعال بروند.

❖ موارد کاربرد:

CAN به طور گسترده در شبکه‌های داخلی خودرو برای ارتباط بین واحدهای کنترل الکترونیکی (ECU)، در سیستم‌های اتوماسیون صنعتی، تجهیزات پزشکی، ماشین‌آلات سنگین و سامانه‌های انرژی به کار می‌رود. همچنین نسخه‌های مبتنی بر CAN مانند CANopen و DeviceNet در صنعت برای استانداردسازی لایه‌های کاربردی توسعه یافته‌اند [10].

۵-۱-۲- استاندارد (USB (Universal Serial Bus

USB یکی از فراگیرترین استانداردهای ارتباطی در سیستم‌های الکترونیکی مدرن است که با هدف یکپارچه‌سازی رابط‌های جانبی و ساده‌سازی اتصال تجهیزات به رایانه توسعه یافت. این استاندارد به دلیل پشتیبانی از سرعت‌های متنوع، قابلیت Plug and Play، مدیریت توان داخلی و ساختار لایه‌بندی‌شده، به یکی از پیچیده‌ترین اما قدرتمندترین پروتکل‌های ارتباطی تبدیل شده است. برخلاف SPI، UART یا I²C که بیشتر برای ارتباطات درون‌بردی یا ساده به کار می‌روند، USB برای ارتباطات بیرونی، انتقال داده با حجم بالا و تعامل مستقیم با سیستم‌های میزبان طراحی شده است. در

سیستم‌های نهفته، USB معمولاً برای ارتباط دستگاه با رایانه، به‌روزرسانی نرم‌افزار، انتقال داده‌های حجیم یا اتصال تجهیزات ذخیره‌سازی استفاده می‌شود.

❖ ساختار پروتکل:

USB دارای معماری لایه‌بندی‌شده شامل لایه فیزیکی (PHY)، لایه پیوند داده و لایه پروتکل است. ارتباط در USB به صورت Host-Device سازمان‌دهی شده و تمام تبادلات داده توسط میزبان کنترل می‌شود. دستگاه‌ها نمی‌توانند به‌صورت مستقل انتقال را آغاز کنند و تنها در پاسخ به درخواست میزبان داده ارسال می‌کنند. داده‌ها در قالب بسته‌هایی شامل Token Packet، Data Packet و Handshake Packet منتقل می‌شوند. در USB 2.0 طول داده می‌تواند تا 512 بایت در هر بسته در حالت High-Speed برسد. همچنین USB از مفهوم Endpoint استفاده می‌کند که هر دستگاه می‌تواند چندین Endpoint برای انواع مختلف انتقال داشته باشد. چهار نوع انتقال اصلی شامل Bulk، Control، Interrupt و Isochronous تعریف شده‌اند که هر یک برای کاربرد خاصی مانند پیکربندی، انتقال داده حجیم، داده‌های دوره‌ای یا داده‌های حساس به زمان طراحی شده‌اند.

❖ نحوه عملکرد لایه فیزیکی:

در لایه فیزیکی، USB از سیگنال‌دهی تفاضلی بر روی دو خط +D و -D استفاده می‌کند که موجب افزایش مقاومت در برابر نویز می‌شود. در نسخه USB 2.0، انتقال داده در حالت High-Speed با نرخ 480 مگابیت بر ثانیه انجام می‌شود و نیازمند کنترل دقیق امپدانس زوج تفاضلی در حدود 90 اهم است. طول مسیرها باید به‌صورت متقارن طراحی شود تا اختلاف فاز و اعوجاج کاهش یابد. همچنین استفاده از حفاظت‌های ESD برای جلوگیری از آسیب ناشی از تخلیه الکترواستاتیکی ضروری است. وجود Pull-up و Pulldown های داخلی در دستگاه‌ها امکان تشخیص نوع سرعت و فرآیند Enumeration را فراهم می‌کند.

❖ روش‌های زمان‌بندی و انتقال داده:

USB یک سیستم زمان‌بندی مبتنی بر فریم دارد که در آن میزبان به‌صورت دوره‌ای باس را اسکن می‌کند. در USB 2.0، هر فریم دارای بازه زمانی 1 میلی‌ثانیه است و در حالت High-Speed به Microframe های 125 میکروثانیه تقسیم می‌شود. تمام انتقال‌ها در این بازه‌های زمانی زمان‌بندی می‌شوند. انتقال داده بر اساس مدل Polling انجام می‌شود، به این معنا که میزبان به‌طور منظم دستگاه‌ها را بررسی می‌کند. در انتقال Isochronous، تأخیر ثابت و تضمین پهنای باند اولویت دارد، در حالی که در انتقال Bulk صحت داده و ارسال مجدد در صورت خطا اهمیت بیشتری دارد. وجود مکانیزم CRC در لایه داده موجب تشخیص خطا و ارسال مجدد بسته‌ها می‌شود.

❖ سرعت:

USB دارای چندین سطح سرعت است که شامل Low Speed با نرخ 1.5 مگابیت بر ثانیه، Full Speed با نرخ 12 مگابیت بر ثانیه و High Speed با نرخ 480 مگابیت بر ثانیه در استاندارد USB 2.0 می‌شود. در نسخه‌های جدیدتر مانند USB 3.x این سرعت به چندین گیگابیت بر ثانیه افزایش یافته است. انتخاب سرعت به نوع دستگاه و نیاز کاربردی بستگی دارد. افزایش سرعت مستلزم طراحی دقیق‌تر مسیرهای PCB و رعایت الزامات سیگنالینگ است.

❖ توپولوژی شبکه:

USB از توپولوژی ستاره‌ای سلسله‌مراتبی استفاده می‌کند. در این ساختار یک Host در رأس قرار دارد و از طریق Hub ها می‌تواند چندین دستگاه را مدیریت کند. هر باس تنها یک میزبان فعال دارد و دستگاه‌ها به صورت زنجیره‌ای یا از طریق هاب‌ها متصل می‌شوند. این ساختار امکان مدیریت متمرکز و تخصیص پهنای باند را فراهم می‌کند، اما برخلاف CAN، ارتباط مستقیم بین دو دستگاه بدون حضور میزبان امکان‌پذیر نیست.

❖ مزایا:

سرعت بالا، پشتیبانی گسترده در سیستم‌های عامل، قابلیت Plug and Play، مدیریت توان داخلی، انتقال داده مطمئن با تشخیص و تصحیح خطا و استانداردسازی جهانی از مهم‌ترین مزایای USB محسوب می‌شوند. همچنین امکان تأمین توان الکتریکی از طریق کابل USB، نیاز به منبع تغذیه جداگانه را در بسیاری از کاربردها حذف می‌کند.

❖ محدودیت‌ها:

پیچیدگی بالای پیاده‌سازی نسبت به پروتکل‌های ساده، نیاز به کنترل دقیق طراحی سخت‌افزاری، وابستگی کامل به میزبان، مصرف توان بالاتر و هزینه بیشتر نسبت به رابط‌های ساده‌تر از جمله محدودیت‌های USB هستند. علاوه بر این، فرآیند توسعه Firmware برای USB به دلیل ساختار لایه‌ای و نیاز به مدیریت Endpoint ها پیچیده‌تر است.

❖ میزان نویز پذیری:

به دلیل استفاده از سیگنال‌دهی تفاضلی و استانداردهای سخت‌گیرانه در طراحی فیزیکی، USB در برابر نویز عملکرد مناسبی دارد، اما در سرعت‌های بالا نسبت به اعوجاج سیگنال، عدم تطابق امپدانس و Crosstalk حساس است. طراحی نادرست مسیرها می‌تواند منجر به کاهش کیفیت سیگنال و خطا در انتقال شود.

❖ موارد کاربرد:

USB در سیستم‌های نهفته برای ارتباط با رایانه، انتقال فایل‌های حجیم، اتصال حافظه‌های فلش، دوربین‌ها، تجهیزات صوتی، کیبورد و ماوس، به‌روزرسانی Firmware و همچنین در نقش واسط سرپال

مجازی (Virtual COM Port) استفاده می‌شود. این استاندارد پلی میان دنیای سیستم‌های نهفته و رایانه‌های شخصی ایجاد کرده و امکان تعامل گسترده میان آن‌ها را فراهم می‌سازد [11].

۶-۱-۲- استاندارد LIN (Local Interconnect Network)

LIN یک استاندارد ارتباطی سریال کم‌هزینه و نسبتاً ساده است که عمدتاً برای شبکه‌های محلی در خودرو و سامانه‌های تعبیه‌شده‌ای طراحی شده که به نرخ داده پایین و پیچیدگی کمتر از CAN نیاز دارند. فلسفه LIN این است که به جای استفاده از یک شبکه پرهزینه و پیچیده برای همه گره‌ها، زیرسیستم‌های غیرحیاتی و کم‌سرعت مانند تجهیزات رفاهی و بدنه خودرو روی یک شبکه ارزان‌تر قرار گیرند و از طریق یک گره دروازه (Gateway) به شبکه‌های اصلی‌تر مانند CAN متصل شوند. به همین دلیل، LIN معمولاً به عنوان «شبکه مکمل CAN» شناخته می‌شود؛ یعنی جایی که الزامات ایمنی، تحمل خطا و زمان‌بندی سخت‌گیرانه CAN لازم نیست، LIN می‌تواند هزینه، سیم‌کشی و سربار پیاده‌سازی را کاهش دهد.

❖ ساختار پروتکل:

LIN یک پروتکل مبتنی بر مدل Master-Slave است و بر خلاف CAN که پیام‌محور و چند-مستر است، در LIN تنها یک Master وجود دارد که زمان‌بندی ارتباط را کنترل می‌کند و Slave‌ها صرفاً در زمان تعیین‌شده پاسخ می‌دهند. تبادل داده در LIN در قالب «فریم» انجام می‌شود که معمولاً از دو بخش اصلی تشکیل می‌گردد: Header که توسط Master ارسال می‌شود و Response که توسط یک Slave (یا در برخی حالت‌ها Master) تولید می‌شود. Header شامل Break (برای همگام‌سازی و تشخیص شروع فریم)، Sync Field (برای کالیبراسیون نرخ بیت و همگام‌سازی دقیق‌تر) و Identifier است که علاوه بر شناسه پیام، بیت‌هایی برای افزایش قابلیت تشخیص خطا نیز دارد. پس از Header، بخش Response شامل ۱ تا ۸ بایت داده و یک Checksum برای بررسی صحت انتقال است. در نسخه‌های جدیدتر LIN، Checksum معمولاً به شکل «Enhanced» تعریف می‌شود تا پوشش بهتری روی داده و شناسه ایجاد کند و احتمال خطای کشف‌نشده کاهش یابد.

❖ نحوه عملکرد لایه فیزیکی:

در لایه فیزیکی، LIN معمولاً به صورت تک‌سیم (Single-Wire) تعریف می‌شود و از یک خط داده به همراه زمین مشترک استفاده می‌کند. سیگنال‌دهی در این شبکه نسبت به پروتکل‌های تفاضلی مانند CAN، حساسیت بیشتری به نویز دارد، اما با توجه به نرخ داده پایین و ماهیت کاربردی آن (زیرسیستم‌های غیرحیاتی)، یک مصالحه اقتصادی محسوب می‌شود. در پیاده‌سازی‌های خودرویی، سطح ولتاژ و ویژگی‌های ترنسیور LIN طوری طراحی می‌شود که بتواند در شرایط الکتریکی خودرو

(نوسانات تغذیه، تداخل‌ها و ترنزینت‌ها) عملکرد قابل قبولی داشته باشد. از دید طراحی سخت‌افزار، انتخاب ترنسیور مناسب، توجه به حفاظت ESD و مدیریت نویز و EMC اهمیت دارد، چون باس LIN اغلب در دسته سیم خودرو عبور می‌کند و در معرض تداخل قرار می‌گیرد.

❖ روش‌های زمان‌بندی و انتقال داده:

LIN یک شبکه زمان‌بندی‌شده (Schedule-Based) است؛ یعنی Master یک جدول زمان‌بندی دارد و بر اساس آن، در بازه‌های مشخص Header هر فریم را منتشر می‌کند. در نتیجه، برخلاف CAN که رقابت و داوری برای دسترسی به باس دارد، در LIN برخورد (Collision) به شکل کلاسیک رخ نمی‌دهد، چون شروع ارسال همواره از Master است و Slave‌ها صرفاً مطابق شناسه تعیین‌شده پاسخ می‌دهند. این روش باعث می‌شود رفتار زمانی شبکه قابل پیش‌بینی‌تر باشد، اما انعطاف‌پذیری آن نسبت به شبکه‌های رویداد محور کمتر می‌شود. برای مدیریت نیازهای متغیر، معمولاً در جدول زمان‌بندی فضاهایی برای فریم‌های رویداد محور یا تشخیصی در نظر گرفته می‌شود، ولی همچنان کنترل نهایی در دست Master باقی می‌ماند. بخش Break و Sync Field نیز نقش کلیدی در زمان‌بندی دارند؛ Break به گیرنده‌ها اعلام می‌کند فریم جدید آغاز شده و Sync به Slave کمک می‌کند نرخ بیت را با Master هم‌تراز کند، چون بسیاری از Slaves از اسیلاتورهای ارزان‌تر با دقت کمتر استفاده می‌کنند.

❖ سرعت:

LIN برای نرخ داده پایین طراحی شده و سرعت آن معمولاً تا حدود ۲۰ کیلوبیت بر ثانیه تعریف می‌شود. این سقف سرعت به‌صورت عملی هم منطقی است، چون در شبکه‌های تک‌سیم با محیط نویزی و با هدف کم‌هزینه بودن، افزایش نرخ داده باعث حساسیت بیشتر به اعوجاج، نویز و خطا می‌شود و هزینه طراحی را بالا می‌برد. بنابراین LIN برای سناریوهایی مناسب است که پیام‌ها کوچک هستند، نرخ به‌روزرسانی بالا لازم نیست و زمان پاسخ در حد ده‌ها میلی‌ثانیه قابل قبول است.

❖ توپولوژی شبکه:

توپولوژی LIN به‌صورت باس تک‌سیم با یک Master و چند Slave است. از منظر معماری سیستم، این ساختار بسیار ساده و کم‌هزینه است و سیم‌کشی را کاهش می‌دهد، اما محدودیت مهم آن وابستگی کل شبکه به Master است؛ یعنی اگر Master دچار مشکل شود یا جدول زمان‌بندی درست اجرا نشود، شبکه عملاً کارایی خود را از دست می‌دهد. همچنین چون باس تک‌سیم و مشترک است، اصول طراحی مسیر و کاهش تداخل در دسته سیم و کانکتورها اهمیت دارد، هرچند الزام‌های آن معمولاً سخت‌گیرانه‌تر از USB پرسرعت نیست.

❖ مزایا:

LIN با کاهش تعداد سیم‌ها و ساده‌سازی پروتکل، هزینه سخت‌افزار و توسعه را پایین می‌آورد و برای گره‌هایی که ساده‌اند و داده کمی دارند بسیار مناسب است. ساختار زمان‌بندی‌شده آن پیش‌بینی‌پذیری

خوبی ایجاد می‌کند و استفاده از Slave‌های ارزان‌تر با اسیلاتورهای کم‌دقت‌تر را ممکن می‌سازد. به همین دلیل در صنعت خودرو برای تجهیزات بدنه و رفاهی که نیازمند نرخ داده بالا یا ایمنی عملکردی شدید نیستند، گزینه‌ای رایج است.

❖ محدودیت‌ها:

محدودیت اصلی LIN سرعت پایین و ظرفیت محدود تبادل داده است. همچنین چون مدل Master-Slave دارد، انعطاف‌پذیری شبکه نسبت به سیستم‌های چند-مستر کمتر است و گلوگاه و نقطه شکست اصلی، Master محسوب می‌شود. علاوه بر این، از نظر تحمل خطا و سازوکارهای مدیریت خطا، LIN در سطحی پایین‌تر از CAN قرار دارد و برای کاربردهای ایمنی‌حساس یا محیط‌های بسیار نویزی که نیازمند قابلیت اطمینان بسیار بالا هستند، معمولاً توصیه نمی‌شود.

❖ میزان نویز پذیری:

به دلیل تک‌سیم بودن و سیگنال‌دهی غیر تفاضلی، LIN ذاتاً نسبت به CAN نویز پذیرتر است. با این حال، نرخ داده پایین، فیلترها و طراحی ترنسیورهای خودرویی، و نیز وجود Checksum کمک می‌کند خطاها در سطح منطقی تا حدی کنترل شوند. در طراحی عملی، رعایت اصول EMC/EMI، انتخاب مسیر عبور مناسب در دسته سیم و استفاده از حفاظت‌های ESD و تحمل ترنزیتهای خودرویی برای افزایش پایداری ضروری است.

❖ موارد کاربرد:

LIN معمولاً در خودرو برای کنترل شیشه‌بالابر، آینه‌ها، قفل‌ها، صندوق، سیستم تهویه، کنترل نورپردازی داخلی و سایر عملگرها و سنسورهای بدنه به کار می‌رود. همچنین در برخی سامانه‌های نهفته غیر خودرویی که به یک باس کم‌هزینه برای چند گره ساده نیاز دارند، می‌تواند گزینه مناسبی باشد، هرچند اکوسیستم اصلی آن همچنان صنعت خودرو است [12].

۲-۲- چالش‌های طراحی مدارهای واسط

پس از بررسی ساختار و عملکرد پروتکل‌های CAN، I²C، SPI، UART و LIN، یکی از مهم‌ترین جنبه‌های عملی در پیاده‌سازی این استانداردها، طراحی صحیح مدار واسط و توجه به محدودیت‌های الکتریکی و فیزیکی است. در بسیاری از پروژه‌های صنعتی، خطاهای ارتباطی نه به دلیل ضعف پروتکل، بلکه به علت طراحی نادرست لایه فیزیکی، مسیرهای PCB یا حفاظت ناکافی رخ می‌دهند. در این بخش، مهم‌ترین چالش‌های مهندسی در طراحی مدارهای ارتباطی بررسی می‌شود تا بتوان معیارهای کاربرد محور برای انتخاب رابط مناسب استخراج کرد.

۲-۲-۱- تطبیق سطح ولتاژ و مبدل‌های سطح (Level Shifting)

در سیستم‌های نهفته مدرن، معمولاً تراشه‌ها با ولتاژهای متفاوتی مانند 3.3V، 1.8V و 5V کار می‌کنند. بسیاری از میکروکنترلرها با ۳.۳ ولت کار می‌کنند، در حالی که برخی سنسورها یا ماژول‌ها ممکن است ۵ ولتی باشند. اتصال مستقیم این دو دامنه ولتاژی می‌تواند باعث آسیب دائمی به ورودی‌ها شود یا منجر به رفتار غیرقابل پیش‌بینی گردد.

در پروتکل‌هایی مانند UART و SPI که سیگنال تک‌پایانه دارند، استفاده از مبدل سطح منطقی (Level Shifter) ضروری است. در I²C به دلیل ماهیت Open-Drain، تطبیق سطح ساده‌تر است و می‌توان از مبدل‌های دوطرفه مبتنی بر MOSFET استفاده کرد. در مقابل، در USB و CAN معمولاً از ترنسیورهای اختصاصی استفاده می‌شود که علاوه بر تطبیق سطح، مشخصات سیگنالینگ استاندارد را نیز تضمین می‌کنند [13].

در کاربردهای صنعتی یا خودرویی، علاوه بر اختلاف سطح منطقی، ممکن است نیاز به ایزولاسیون گالوانیکی نیز وجود داشته باشد تا از انتقال نویز، گراند لوپ یا آسیب ناشی از اختلاف پتانسیل زمین جلوگیری شود. در این شرایط از ایزولاتورهای دیجیتال یا ایتوکوپلرها استفاده می‌شود، به‌ویژه در CAN صنعتی و RS-485.

۲-۲-۲- یکپارچگی سیگنال (Signal Integrity)

با افزایش سرعت انتقال داده، مسائل مرتبط با یکپارچگی سیگنال اهمیت بیشتری پیدا می‌کند. در سرعت‌های پایین مانند LIN یا UART کم‌سرعت، اثرات انتقال موج چندان بحرانی نیستند، اما در SPI با چندین مگاهرتز یا USB با صدها مگابیت بر ثانیه، مسیر PCB عملاً مانند یک خط انتقال رفتار می‌کند.

یکی از مشکلات رایج، بازتاب سیگنال (Reflection) است که در صورت عدم تطابق امپدانس مسیر با بار رخ می‌دهد. در CAN، وجود مقاومت‌های انتهایی ۱۲۰ اهم برای جلوگیری از بازتاب ضروری است. در USB 2.0 نیز کنترل امپدانس زوج تفاضلی در حدود ۹۰ اهم اهمیت حیاتی دارد.

پدیده‌های Overshoot و Undershoot نیز می‌توانند موجب عبور سیگنال از محدوده مجاز ولتاژ شوند و در بلندمدت به آسیب ورودی‌ها منجر شوند. این مشکل معمولاً ناشی از لبه‌های تیز سیگنال، طول زیاد مسیر یا طراحی نادرست زمین مرجع است.

اثر Crosstalk بین مسیرهای مجاور نیز در طراحی بردهای چند لایه اهمیت دارد، به‌ویژه زمانی که خطوط کلاک سریع مانند SCLK در SPI در کنار خطوط حساس قرار می‌گیرند. رعایت فاصله مناسب، استفاده از صفحه زمین پیوسته و کنترل طول مسیرها از اصول کلیدی طراحی محسوب می‌شود.

در USB و CAN که از سیگنال‌دهی تفاضلی استفاده می‌کنند، تطابق طول دو مسیر تفاضلی بسیار مهم است، زیرا اختلاف طول موجب ایجاد خطای فاز و کاهش کیفیت سیگنال می‌شود [14].

۳-۲-۲- حفاظت در برابر ESD و Surge

پورت‌های بیرونی مانند LIN، CAN، USB و RS-232 در معرض تخلیه الکترواستاتیکی (ESD) و نوسانات ناگهانی ولتاژ (Surge) قرار دارند. لمس کانکتور توسط کاربر یا اتصال کابل در محیط صنعتی می‌تواند تخلیه‌هایی در حد چند کیلو ولت ایجاد کند. بدون حفاظت مناسب، این پدیده می‌تواند باعث تخریب دائمی ترنسیور شود.

برای حفاظت، معمولاً از دیودهای TVS با ظرفیت خازنی پایین استفاده می‌شود تا بدون تأثیر منفی بر شکل موج سیگنال، اضافه ولتاژ را به زمین هدایت کنند. در استانداردهایی مانند USB که سرعت بالا دارند، انتخاب قطعه حفاظتی با ظرفیت خازنی کم اهمیت ویژه‌ای دارد، زیرا ظرفیت زیاد می‌تواند موجب تضعیف سیگنال شود.

در محیط‌های صنعتی یا خودرویی، حفاظت در برابر Surge و Load Dump نیز اهمیت دارد و طراحی باید مطابق استانداردهای EMC انجام شود. در CAN خودرویی، ترنسیورها معمولاً دارای حفاظت داخلی در برابر ترنزینت‌های ولتاژ هستند [15].

۴-۲-۲- استخراج معیارهای انتخاب رابط مناسب

با در نظر گرفتن چالش‌های طراحی سخت‌افزاری، انتخاب استاندارد ارتباطی صرفاً بر اساس سرعت اسمی انجام نمی‌شود، بلکه باید مجموعه‌ای از عوامل مهندسی ارزیابی شود.

اگر سرعت بالا و انتقال داده حجیم مورد نیاز باشد و طراحی PCB پیشرفته قابل انجام باشد، USB یا SPI انتخاب مناسب‌تری هستند. اگر فاصله انتقال زیاد و محیط نویزی وجود داشته باشد، CAN به دلیل سیگنال‌دهی تفاضلی و مدیریت خطای پیشرفته برتری دارد. در کاربردهای کم‌هزینه و کم‌سرعت با

چند گره ساده، LIN یا I²C انتخاب منطقی‌تری خواهند بود. برای ارتباطات ساده نقطه‌به‌نقطه و دیباگ، UART همچنان کم‌هزینه‌ترین گزینه است.

از نظر نویزپذیری، پروتکل‌های تفاضلی مانند CAN و USB عملکرد بهتری نسبت به پروتکل‌های تک‌پایانه مانند SPI، UART، و I²C دارند. از نظر توپولوژی، اگر شبکه چندگره‌ای مورد نیاز باشد، CAN و I²C مناسب‌تر هستند، در حالی که SPI و UART عمدتاً نقطه‌به‌نقطه هستند. از نظر هزینه پیاده‌سازی، UART و I²C ساده‌ترین و ارزان‌ترین راهکارها را ارائه می‌دهند، در حالی که USB به دلیل پیچیدگی لایه فیزیکی و نرم‌افزاری هزینه بیشتری دارد.

۳- جمع‌بندی

استانداردهای ارتباطی در سیستم‌های نهفته هر یک با هدف پاسخ‌گویی به نیازهای خاصی توسعه یافته‌اند و هیچ پروتکلی به‌تنهایی پاسخگوی تمام سناریوهای طراحی نیست. UART با سادگی و هزینه پایین، گزینه‌ای مناسب برای ارتباطات نقطه‌به‌نقطه و کاربردهای دیباگ است، در حالی که SPI با سرعت بالا برای تبادل داده درون‌بردی به‌کار می‌رود. I²C با کاهش تعداد خطوط و قابلیت آدرس‌دهی چندگانه، راهکاری متعادل برای اتصال سنسورها و تراشه‌های جانبی فراهم می‌کند. در سطح شبکه‌های مقاوم و بلادرنگ، CAN به دلیل سیگنال‌دهی تفاضلی، داوری غیرمخرب و سازوکارهای پیشرفته مدیریت خطا، انتخابی قابل اعتماد برای محیط‌های صنعتی و خودرویی محسوب می‌شود و LIN به عنوان نسخه‌ای کم‌هزینه برای زیرسیستم‌های کم‌سرعت مکمل آن است. USB نیز با ساختار لایه‌بندی‌شده و سرعت بالا، ارتباط میان سیستم‌های نهفته و تجهیزات خارجی را ممکن می‌سازد. با این حال، عملکرد واقعی این پروتکل‌ها به شدت به کیفیت طراحی لایه فیزیکی، تطبیق سطح ولتاژ، کنترل یکپارچگی سیگنال، رعایت اصول مسیریابی PCB و حفاظت در برابر ESD و نویز وابسته است. بنابراین انتخاب رابط مناسب باید بر اساس مجموعه‌ای از معیارهای کاربرد محور شامل سرعت مورد نیاز، فاصله انتقال، شرایط نویزی محیط، توپولوژی شبکه، سطح اطمینان مورد انتظار و هزینه پیاده‌سازی انجام گیرد. در نهایت، رویکرد مهندسی صحیح مستلزم تحلیل همزمان ویژگی‌های پروتکل و ملاحظات عملی سخت‌افزاری است تا سامانه‌ای پایدار، ایمن و بهینه در کاربرد مورد نظر حاصل شود.

۴- مراجع

[1]https://www.ti.com/lit/ug/sprugp1/sprugp1.pdf?utm_source=chatgpt.com&ts=1770632877009&ref_url

- [2]https://www.analog.com/en/resources/analog-dialogue/articles/introduction-to-spi-interface.html?utm_source
- [3]https://www.nxp.com/docs/en/user-guide/UM10204.pdf?utm_source
- [4]https://www.iso.org/standard/63648.html?utm_source
- [5]https://www.cs-group.de/wp-content/uploads/2016/11/LIN_Specification_Package_2.2A.pdf?utm_source
- [6]https://www.usb.org/document-library/usb-20-specification?utm_source
- [7]<https://www.ti.com/lit/an/slla037a/slla037a.pdf>
- [8]<https://www.ti.com/lit/ug/sprugp2a/sprugp2a.pdf>
- [9]https://www.nxp.com/docs/en/user-guide/UM10204.pdf?utm_source=chatgpt.com
- [10]<https://www.ti.com/lit/an/sloa101b/sloa101b.pdf>
- [11]<https://www.beyondlogic.org/usbnutshell/usb1.shtml>
- [12]https://www.cs-group.de/wp-content/uploads/2016/11/LIN_Specification_Package_2.2A.pdf?utm_source=chatgpt.com
- [13]<https://www.ti.com/lit/an/scea044/scea044.pdf>
- [14]<https://www.ti.com/lit/an/slla414a/slla414a.pdf?ts=1771239207384>
- [15]<https://www.ti.com/lit/an/slla302/slla302.pdf>