

زمان بندی حلقه

Loop scheduling

این مقاله سعی در پاسخ گویی به سوالات زیر دارد:

چه چیزی تعیین میکند که یک function block یا مخابرات شرع بشود؟

چگونه از همپوشی مخابرات (communication overlap) جلوگیری میشود، وقتی که بیشتر از یک حلقه بر رویه یک segment وجود دارد؟

چگونه با حلقه های کند و تند وقف داده میشود؟

در این مقاله می خواهیم در باره نحوه زمان بندی در حلقه در یک segment فیلدباس بحث نماییم. ابتدا به سؤال چه عاملی باعث انجام عملیات کنترلی در زمانی که می بایست انجام پذیرد میگردد و سپس زمان بندی پایه و همچنین بررسی حلقه های متعدد رویه یک segment را بررسی میکنیم.

مقدمه

شروع می کنیم: با سوال "چه عاملی باعث انجام عملیات در زمان مطلوب می باشد؟"

چه عاملی باعث انجام عملیات ها در زمان مطلوب است؟

کنترل فرایند خوب وابسته به زمان می باشد. اگر عملیات های کنترلی، زمانی که باید انجام شوند اجرا نشوند، تغییرات فرایندی بوجود آمده می توانند باعث بالا رفتن مصرف انرژی و مواد خام، کم شدن محصول و پایین آمدن کیفیت کالا شود.

FOUNDATION fieldbus این مشکل را با اجرای کنترل در یک زمان بندی deterministic و real-time حل می کند. این تکنولوژی طراحی شده است تا بتواند یک بازه کامل از موقعیت های احتمالی کنترلی را پوشش دهد.

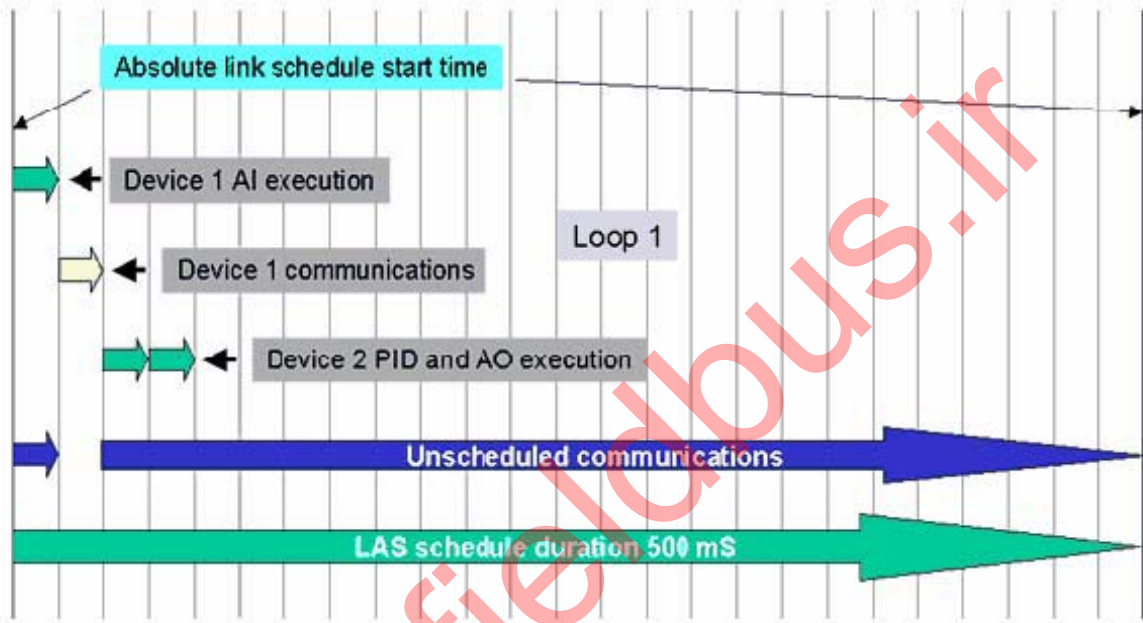
زمان بندی پایه

در FOUNDATION fieldbus، ارتباطات مخصوص کنترل و function block ها در فواصله دقیقاً تعیین شده و با ترتیب زمان بندی شده برای کنترل فرایندی صحیح، اجرا میشوند.

روش های اجرای صحیح زمان بندی ارتباطات در fieldbus شامل application clock و link active scheduler است. در این مقاله چگونگی سنکرون کردن آنها و function block ها را برای رسیدن به مطلوب ترین کارایی را بحث خواهیم کرد.

زمان بندی کلی، **macrocycle** نام دارد. **macrocycle** برای تمامی ابزار های رویه یک **segment** دقیقاً زمان بندی شده و همه از یک زمان شروع استفاده میکنند. و ارتباطات در یک فاصله زمانی از زمان شروع اجرا میشوند.

شکل زیر زمان بندی یک حلقه تیپ را در حالی که **PID function** در شیر (Device 2) نصب شده است را نشان میدهد. هر فعالیت در فاصله مشخصی از زمان شروع مطلق رخ میدهد.

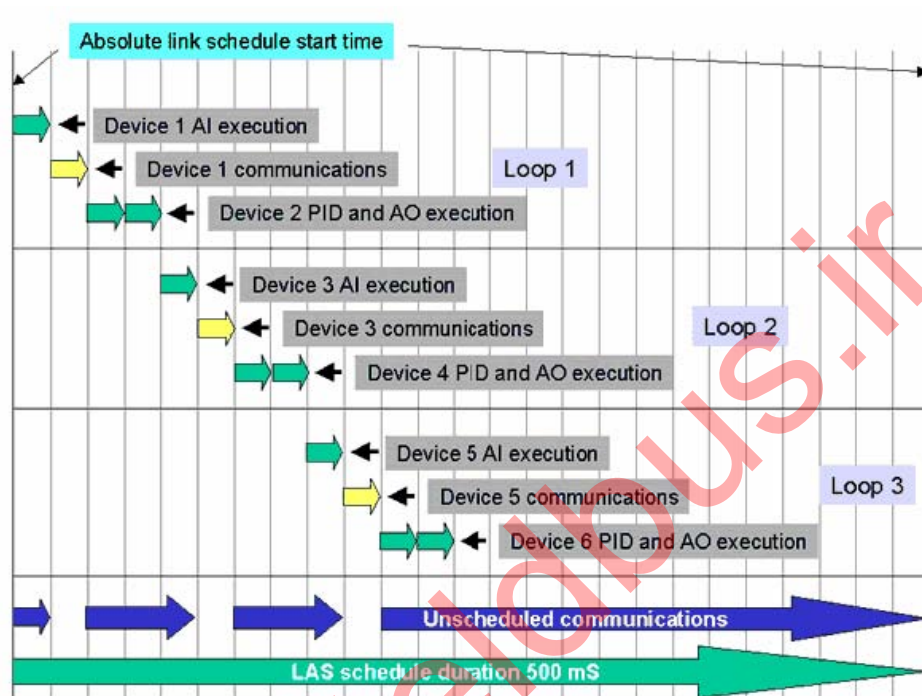


زمان بندی تیپ

این سیکل در یک زمان بندی دقیق و مداوم تکرار میشود. پیغام های زمان بندی نشده (**acyclic**) میتوانند در هر زمانی که بسته های زمان بندی شده (**cyclic**) ارسال نمی شوند، ارسال شوند.

نکته کاربردی

در زمان بندی حلقه ها دقت کنید. توابع با ترتیبی که شما تعیین میکنید اجرا میشوند، حتا اگر این ترتیب غلط باشد. مثلاً در زمان بندی، اجرای **AO** در اول و **PID** بعد از آن و **AI** در درجه آخر، باعث اضافه شدن یک تاخیر بزرگ و بی دلیل در حلقه میشود.



زمان بندی حلقه های چندگانه

همان طور که این مثال نشان میدهد، شما میتوانید چندین **function block** را در یک زمان و بر روی یک **segment** در حال اجرا داشته باشید، به شرطی که آن ها در تجهیزات مختلف باشند و زمان های شروع متفاوت داشته باشند. مثال بالا ۳ حلقه دارد، با **PID** ها در شیر های کنترلی. البته بیشتر از یک تجهیز نمیتواند بر رویه یک **bus** و در یک زمان ارسال اطلاعات کند. این زمان بندی باعث جلوگیری از همپوشی ارتباطات (**communication overlap**)، با پلکانی کردن زمان شروع ارسال میشود، تا یک **block** قبل از اتمام یافتن **block** قبل شروع نشود.

به خاطر ساده کرده موضوع، این نمودار، اجرای **block** ها را زنجیر وار نشان داده است با هیچ همپوشی پردازش. در واقعیت، چندین **block** میتوانند در یک زمان اجرا شوند تا وقتی که آنها در تجهیزات مختلف باشند و اطلاعات بتواند به محض این که پردازش تمام شود مکالمه بشوند. چندین تجهیز نمیتوانند در یک زمان مخابره کنند.

نکته کاربردی

فرض نکنید همه تجهیزات یک سطح کارایی دارند. تجهیزات مختلف ممکن است نیاز به زمان های مختلف برای اجرای یک **function block** داشته باشند. برای مثال یک تجهیز ممکن است یک **PID block** را در ۳۰ms اجرا کند و دیگری در ۷۵ms. بررسی کنید تا ببینید آیا فروشنده میتواند ابزار زمان بندی اتوماتیکی (**automated scheduling tool**) در اختیار شما بگذارد که زمان های مختلف اجرا را مد نظر بگیرد.

www.fielabus.ir