

تفاوت روش سرکشی و وقفه:

در روش سرکشی، CPU/پردازنده (به صورت نرم افزاری) در یک حلقه بی نهایت مدام چک میکند که آیا از طرف دستگاه های I/O پاسخ آمده یا خیر و در این حالت CPU اشغال است ولی کار بیهوده انجام میدهد.

ولی در روش وقفه، یک مکانیزم است که پردازنده را در تعامل با دستگاه های I/O قادر میسازد که منتظر نماند بلکه به کار خود در صورت وجود ادامه میدهد و زمانی که I/O پاسخ داد از طریق وقفه به CPU اطلاع میدهد.

پرسش: چرا این روش برای فهمیدن اینکه چه زمانی کلید بسته شده درست نیست؟ در این مدار پایه میکرو در چه حالتی می باشد؟

اگر مقاومت در مدار نباشد، زمانی که switch وصل بشود مانند آنتن عمل کرده، باعث جریان ایجاد قوی در مدار شده و جریان زیادی به پایه میکرو وارد میشود و نویز در مدار ایجاد میشود و مقادیر پایه به اشتباه تغییر میکند و مقدار دقیق روی پین نداریم .

پایه های میکرو در حالت floor قرار میگیرد.

پرسش: درباره چگونگی کارکرد مدارهای بالا توضیح دهید. به چه دلیل نیاز به مقاومت (Pull-up/pull-Down) داریم؟

برای جلوگیری از ورود نویز و آسیب دیدن سیستم و کنترل جریان ورودی به دستگاه برای کنترل توان مصرفی از Pull-up/pull-Down استفاده می کنیم. در حالت Pull-Up اگر کلید بسته بشود، جریان از VCC به زمین داریم و مقدار پایه، صفر منطقی است. ولی اگر باز باشد، VCC به پایه میکروکنترلر وصل است و مقدار یک را برمی گرداند. در مثنای Pull-Down اگر سوئیچ باز باشد، در نتیجه پایه به زمین وصل است و مقدار صفر دارد و وقتی که بسته باشد، به VCC وصل است و یک می شود.

پرسش: آیا رخ دادن یک اتفاق در صورت اعلام شدن (Assertion) لزوماً منجر به اجرای روال سرویس وقفه متناظر با آن می شود؟

خیر. با توجه به نحوه پیاده سازی دستورها و اولویت بندی در پردازنده، اگر همزمان با اعلام شدن وقفه سرویس دیگری در حال انجام باشد، پردازنده باید ابتدا instruction های سرویس کنونی را به اتمام برساند تا دیگری انجام شود.

پرسش: پایه‌های وقفه در برد ATmega 2560 و شیوه پیاده‌سازی وقفه ورودی را به دست آورید.

پایه‌های 2، 3، 18، 19، 20 و 21 مختص به وقفه‌ها هستند و با استفاده از تابع attachInterrupt به شماره port پایه‌های میکروی مورد نظر دستورات ISR مان، متصل می‌کنیم.

پرسش: اگر بخواهیم در زمان تغییر مقدار پایه، وقفه فعال شود از چه mode ای درون تابع attachInterrupt استفاده می‌شود؟

باید از Change استفاده کنیم.

پرسش: انواع اتفاق های ورودی را که واحد GPIO در برد آردوینو ATmega 2560 می‌تواند رخ دادن آن‌ها را بفهمد و اعلام کند بنویسید.

Falling – Rising – Change – Low

اگر دکمه را در حالت فشرده برای زمان طولانی نگه داریم چه اتفاقی خواهد افتاد؟ آیا با منطق کارکرد خواسته شده سازگار است؟ چه راه حلی برای این مشکل (در صورت وجود) می‌توان پیشنهاد کرد؟

سرکشی: عملیات تعریف شده برای دکمه‌ها مدام تکرار می‌شود و با منطق کارکرد خواسته شده، سازگار نیست و می‌توانیم یک متغیر Boolean تعریف کنیم تا این مشکل حل شود.

وقفه: کارکرد برنامه درست است زیرا در حالت وقفه Change است.

فرض کنید می‌خواهیم کارکرد دیگری را به دستگاه اضافه کنیم به این صورت که در صورت یک شدن یک پایه عملیات مشخصی را به عنوان پاسخ انجام دهد. (محدودیت زمانی برای پاسخ دادن وجود دارد) هیچ یک از اتفاق‌های یک شدن پایه نباید از دست برود (بی پاسخ بماند). و یک شدن پایه نیز در هر زمانی ممکن است رخ دهد. آیا برنامه شما که به روش سرکشی واحد های جانبی را بررسی می‌تواند در هر شرایطی (مثلاً هنگام فشرده شدن کلید) این کارکرد را فراهم کند؟

سرکشی: سیستم هنگام Delay دستور نمی‌خواند اما پس از هر Delay می‌توان تابع مورد نظر را چک کرد و در صورت درست بودن شرط را انجام داد. در نتیجه نمی‌شود.

وقفه: سیستم در هنگام وقفه کار دیگری را انجام نمی‌دهد. در نتیجه نمی‌شود.

فرض کنید به دلیل محدودیت در توان مصرفی می خواهیم پردازنده در هنگام بیکاری به خواب برود. در زمان خواب پردازنده هیچ دستوری را اجرا نمی کند. روش سرکشی چه قدر با این نیازمندی سازگاری دارد؟ آیا می توان با این روش هم به خواب رفت و هم کارکرد درست آزمایش را فراهم کرد؟

سرکشی: در تابع loop در هر بار شرطهای دکمه ها چک می شود و اگر سیستم به خواب برود چک نمی شود. در نتیجه نمی شود. وقفه: در حالت deepsleep سیستم نسبت به پایه های وقفه حساس است و اگر جریان برقرار شود سیستم بخ حالت روشن برمی گردد.