

به نام خدا

پیش گزارش آزمایش اول آزمایشگاه ریز پردازنده

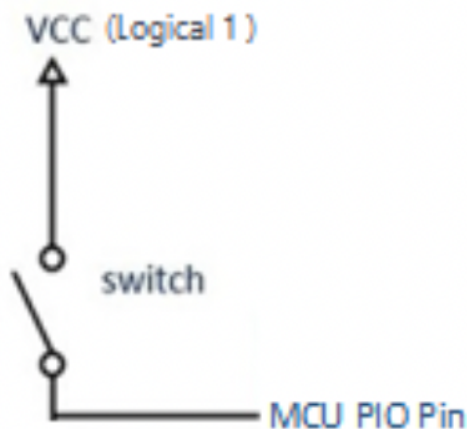
امیرپارسا سلمان خواه

۹۸۳۱۰۳۴

تفاوت های روش سرکشی و وقفه محور:

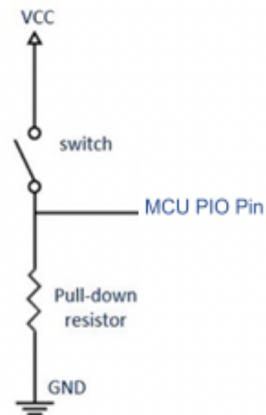
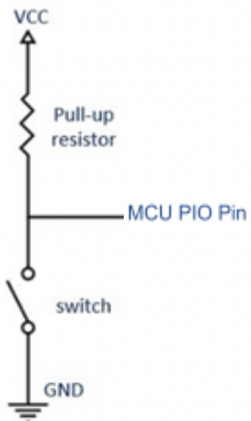
در روش سرکشی مقدار یک flag در پردازنده به طور پی در پی چک می شود و در صورتی که تغییری در مقدار آن ایجاد شود (مثلا مقدار آن ۱ شود)، پردازنده از اتمام عمل خواندن یا نوشتن روی دستگاه های IO آگاه می شود و سراغ ادامه ی کار می رود. اما در روش وقفه محور، Device Controller ها وظیفه ی انتقال داده ها از حافظه ی اصلی به دستگاه های IO و یا برعکس را به عهده می گیرند و بعد از اتمام کار با ارسال یک وقفه، پردازنده را از اتمام فرایند آگاه می کنند. پردازنده در این روش در حین انجام فرایند انتقال می تواند به سایر کار های خود رسیدگی کند.

پرسش ۱: چرا روش زیر برای فهمیدن اینکه کلید چه زمانی بسته شده است درست نیست؟ در این مدار پایه ی ماکرو در چه حالتی قرار دارد؟



در حالتی که کلید در حالت باز قرار دارد پایه ی ماکرو به جایی متصل نیست و در حالت float قرار دارد. در این حالت پایه ی مدار در یک حالت ناپایدار قرار می گیرد و مقادیر معتبری از خود نشان نمی دهد و ممکن است بین ۰ و ۱ نوسان کند و به اشتباه مقدار ۱ را به پایه ی ماکرو برگرداند. به همین دلیل از مقاومت های pull-up و pull-down استفاده می شود که در پرسش های بعدی به توضیح آن ها می پردازم.

پرسش ۲: درباره ی چگونگی کارکرد مدار های زیر توضیح دهید. به چه دلیل نیاز به مقاومت های Pull-up یا Pull-down داریم؟



در مدار سمت راست که مقاومت Pull-down به مدار اضافه شده، هنگامی که کلید باز باشد، از آن جا که مقدار جریان عبوری از پایه های ورودی ماکرو بسیار کم و نزدیک به صفر است، اختلاف پتانسیل بین زمین و پایه ی ماکرو برابر با حاصل ضرب جریان مدار در مقاومت pull-down است که این مقدار به دلیل کم بودن جریان نزدیک به صفر خواهد بود. بنابراین ولتاژ ورودی پایه ی ماکرو صفر خواهد بود و مقدار ۰ منطقی را خواهد داشت. در حالتی که کلید بسته است هم مانند مدار قبلی، به دلیل آن که پایه ی ماکرو به VCC متصل است، مقدارش برابر با VCC خواهد بود.

در مدار سمت چپ که مقاومت Pull-up به مدار اضافه شده، هنگامی که کلید باز باشد، از آن جا که مقدار جریان عبوری از پایه های ورودی ماکرو بسیار کم و نزدیک به صفر است، اختلاف پتانسیل بین VCC و پایه ی ماکرو برابر با حاصل ضرب جریان مدار در مقاومت pull-down است که این مقدار به دلیل کم بودن جریان نزدیک به صفر خواهد بود. بنابراین ولتاژ ورودی پایه ی ماکرو بسیار نزدیک به VCC خواهد بود و مقدار ۱ منطقی را خواهد داشت. در حالتی که کلید بسته است از آن جا که پایه ی ماکرو به زمین متصل است مقدار ۰ منطقی را خواهد داشت.

بنابراین با استفاده از این مقاومت ها می توانیم به درستی ورودی کاربر را به پایه های ماکرو منتقل کنیم و در هر دو حالت ۰ و ۱ از اعتبار این ورودی اطمینان داریم.

پرسش ۳: آیا رخ دادن یک اتفاق در صورت اعلام شدن لزوما منجر به اجرای روال سرویس وقفه ی متناظر با آن می شود؟

خیر. پردازنده برای پردازش وقفه های تو در تو، آستانه ای دارد و ممکن است نتواند چندین وقفه ی داخل هم را پشتیبانی کند. به عنوان مثال در آردوینوی uno فقط یک وقفه ی تو در تو پشتیبانی میشود و اگر دو یا چند وقفه در هنگام اجرای روال سرویس یک وقفه ی دیگر اتفاق بیفتند روال سرویس آن ها اجرا نخواهد شد.

پرسش ۴: پایه های وقفه در برد Atmega 2560 و شیوه ی پیاده سازی وقفه ی ورودی را به دست آورید.

با توجه به جدول زیر که از سایت آردوینو استخراج شده است، پایه های ۲۰، ۱۹، ۱۸، ۳، ۲ و ۲۱ در این برد برای پیاده سازی وقفه استفاده می شوند.

BOARD	DIGITAL PINS USABLE FOR INTERRUPTS
Uno, Nano, Mini, other 328-based	2, 3
Uno WiFi Rev.2, Nano Every	all digital pins
Mega, Mega2560, MegaADK	2, 3, 18, 19, 20, 21 (pins 20 & 21 are not available to use for interrupts while they are used for I2C communication)
Micro, Leonardo, other 32u4-based	0, 1, 2, 3, 7
Zero	all digital pins, except 4
MKR Family boards	0, 1, 4, 5, 6, 7, 8, 9, A1, A2
Nano 33 IoT	2, 3, 9, 10, 11, 13, A1, A5, A7
Nano 33 BLE, Nano 33 BLE Sense	all pins
Due	all digital pins
101	all digital pins (Only pins 2, 5, 7, 8, 10, 11, 12, 13 work with CHANGE)

برای پیاده سازی وقفه از تابع attachInterrupt استفاده می شود. این تابع شماره ی پایه ی ورودی، تابع روال سرویس وقفه و رویداد اجرای وقفه را به عنوان ورودی دریافت می کند.

برای دادن شماره ی پایه ی ورودی از تابع `digitalPinToInterrupt` استفاده می شود که آن هم شماره ی پایه ی ورودی را به عنوان وردی دریافت می کند. تابع `روال سرویس وقفه` نیز یک تابع معمولاً کوتاه و بدون آرگومان ورودی است که نام این تابع به عنوان ورودی به `attachInterrupt` پاس داده می شود. رویداد اجرای وقفه هم در واقع اتفاقی است که لازم است روی پایه ی ورودی بیفتد تا `روال سرویس صدا زده` شود که در ادامه ی پرسش ها به توضیح انواع مختلف آن ها می پردازم.

پرسش ۵: اگر بخواهیم در هنگام تغییر مقدار پایه، وقفه فعال شود از چه mode ای در درون تابع `attachInterrupt` استفاده می شود؟

از حالت `CHANGE` برای این کار استفاده می شود.

پرسش ۶: انواع اتفاق های ورودی را که واحد `GPIO` در برد آردوینو `Atmega ۲۵۶۰` می تواند رخ دادن آن ها را بفهمد و اعلام کند بنویسید.

با توجه به سایت آردوینو، این برد می تواند ۴ حالت زیر را متوجه شود.

- **LOW** to trigger the interrupt whenever the pin is low,
- **CHANGE** to trigger the interrupt whenever the pin changes value
- **RISING** to trigger when the pin goes from low to high,
- **FALLING** for when the pin goes from high to low.

حالت اول برای زمانی که پایه ی ورودی در حالت ۰ منطقی است.
حالت دوم برای زمانی که پایه ی ورودی در حال تغییر است.
حالت سوم برای زمانی که پایه ی ورودی از حالت ۰ به حالت ۱ می رود.
حالت چهارم برای زمانی که پایه ی ورودی از حالت ۱ به حالت ۰ می رود.