

آزمایش ۴

آزمایشگاه ریزپردازنده
نیم سال دوم ۱۴۰۱-۱۴۰۰

هدف

هدف از این آزمایش آشنایی با تایمر / شمارنده و شیوه راه اندازی و به کارگیری آن‌ها در میکروکنترلر STM32F401 است.

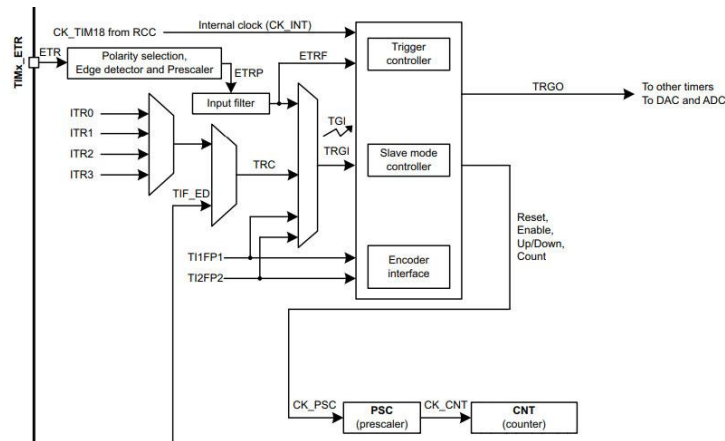
پیش نیاز و مطالعه

- آشنایی با مفاهیم تایمر

به عنوان یک تعریف ساده و کلی، تایمر ابزاری است که زمان را می‌سنجد، و سنجش زمان می‌تواند با به کارگیری سیستم‌ها و روش‌های مختلفی انجام شود. ساخت تایمر بر پایه یک شمارنده است. برای سنجش زمان، یک شمارنده داریم که عمل شمارش را بر اساس یک کلاک متناوب انجام می‌دهد و برای محاسبه‌ی زمان، باید عدد شمارش شده توسط شمارنده را در مدت زمان هر پالس ساعت ضرب کنیم. فرض کنید یک شمارنده ۴ بیتی داریم که کلاک این شمارنده از یک منبع کلاک با فرکانس ۱ KHz تأمین می‌شود. این شمارنده ۴ بیتی می‌تواند از عدد ۰ تا ۱۵ را شمارش کند و با توجه به فرکانس ۱ KHz، میزان هر شمارش ۱ ms است. پس با به کارگیری یک شمارنده می‌توان تایمری ساخت که می‌تواند زمان را با پایه زمانی ۱ ms بسنجد و برای زمان‌های دیگر هم می‌توان عدد ۱ تا ۱۶ (با احتساب عدد ۰ به عنوان اولین عدد شمارش) را در ۱ ms ضرب کرد. عموماً تایمر در میکروکنترلرهای STM32 دارای یک شمارنده ۱۶ بیتی است (در سری‌های متفاوت این عدد تا ۳۲ بیت هم می‌رسد) که می‌تواند به صورت بالا شمار، پایین شمار و بالا-پایین شمار، شمارش کند. نکته مهم این است که این شمارنده ۱۶ بیتی با چه فرکانسی شمارش می‌کند.

تایمر در میکروکنترلرهای STM32 دارای یک Prescaler با طول ۱۶ بیت است که فرکانس ورودی واحد تایمر را به عددی بین ۱ تا ۶۵۵۳۶ تقسیم می‌کند. پس علاوه بر اینکه با به کارگیری Prescaler ها و تقسیم‌کننده‌های فرکانسی که پیش از واحد تایمر قرار دارند، می‌توان فرکانس ورودی واحد تایمر را تعیین کرد، با به کارگیری Prescaler که در خود واحد تایمر قرار دارد هم این انعطاف موجود است که فرکانس را تا حد بسیار زیادی، و تقریباً به هر عدد دلخواه تغییر داد.

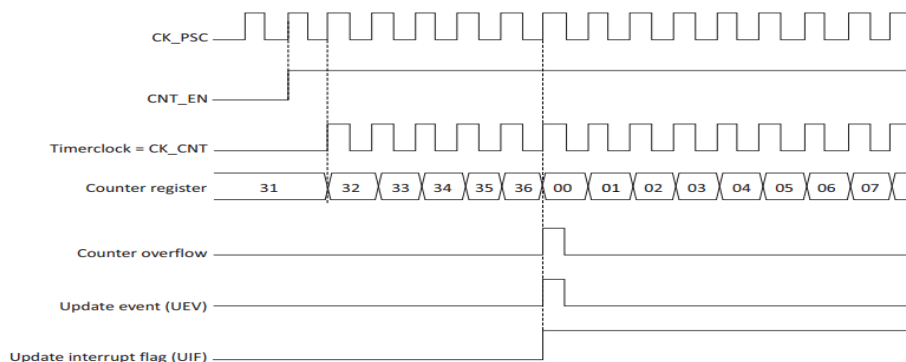
هم Prescaler و هم شمارنده هر دو ۱۶ بیتی هستند و Prescaler دقیقاً قبل از شمارنده قرار داده شده است و کلاک متصل به شمارنده، همان کلاکی است که از خروجی Prescaler گرفته می‌شود. کلاک ورودی به واحد تایمر در میکروکنترلرهای STM32 می‌تواند از روش‌های مختلفی مانند کلاک داخلی میکروکنترلر، پین‌های خارجی و از طریق یک تایمر دیگر تأمین بشود.



همانطور که در تصویر بالا مشاهده می‌شود، کلاک از هر منبعی که تأمین بشود، قبل از رفتن به شمارنده، ابتدا توسط Prescaler تقسیم فرکانسی می‌شود و در پایان به شمارنده متصل خواهد شد.

در ادامه می‌خواهیم با به کارگیری تایمر، قابلیت پایه زمانی را راه‌اندازی و مدت زمان ۱ ثانیه را اندازه‌گیری بکنیم. تایمر را در حالت بالا شمار راه‌اندازی کرده و پس از اینکه اعداد شمارش شده توسط شمارنده برابر با ۱ ثانیه شد، وقفه واحد تایمر را فعال کرده تا متوجه سپری شدن زمان ۱ ثانیه بشویم و متناسب با آن عملیات موردنظر را انجام بدهیم.

اما چگونه می‌توان متوجه شد که اعداد شمارش شده توسط شمارنده، معادل ۱ ثانیه است؟ با به کارگیری منبع کلاک و Prescaler، کلاک ورودی شمارنده واحد تایمر مشخص می‌شود. به عنوان نمونه منبع کلاک را از نوع داخلی انتخاب کرده و مقدار آن را بر روی ۸ MHz و Prescaler را بر روی ۷۹۹۹ (به این دلیل ۸۰۰۰ قرار ندادیم چون که شمارش از ۰ آغاز می‌شود) تنظیم می‌شود. با این تنظیمات منبع کلاک بر عدد Prescaler بخش شده و در پایان به ورودی کلاک شمارنده متصل می‌شود. یعنی ۸MHz بر عدد ۸۰۰۰ تقسیم شده و در پایان فرکانس ۱ KHz به ورودی کلاک شمارنده متصل می‌شود. با این تنظیمات با اضافه شدن هر عدد به شمارنده، مدت زمان ۱ms سپری می‌شود. حال باید شمارنده تا عدد ۱۰۰۰ بشمارد تا بتوان مدت زمان ۱ ثانیه را به دست آورد. برای این منظور یک رجیستر به نام ARR یا Auto-reload وجود دارد و زمانی که مقدار شمارنده به مقدار این رجیستر رسید، تایمر یک وقفه تولید می‌کند و سپری شدن زمان مدنظر را اعلام می‌کند تا عملیات دلخواه در روتین وقفه رجیستر شده انجام شود.



نحوه‌ی ایجاد وقفه پس از رسیدن مقدار شمارنده به رجیستر Auto-reload

در تصویر بالا مقدار رجیستر Auto-reload عدد ۳۶ است و زمانی که مقدار شمارنده به عدد ۳۶ برسد شمارنده ریست و یک وقفه (Update interrupt flag) هم ایجاد خواهد شد.

همچنین توجه کنید که شمارنده یک کلاک پس از فعال شدن کلاک ورودی شمارنده (CNT_EN) شروع به شمارش می‌کند. این یعنی اگر قرار باشد تا ۱۰۰۰ بشماریم باید مقدار Auto-reload را عدد ۹۹۹ تنظیم کنیم چرا که عدد ۰ هم با توضیحی که اکنون داده شد شمارش خواهد شد.

سؤالات تحلیلی

۱. مدهای عملیاتی تایمر را در میکروکنترلر stm32f4 نام ببرید. هر یک از این مدها چه کاربردی دارد؟
۲. تایمر RTC (Real-Time Clock) و کاربرد اصلی آن چیست؟ تایمر Watchdog و کاربردی آن چیست؟
۳. منظور از قابلیت Repetition شمارنده در برخی از میکروکنترلرها چیست؟

دستور کار

۱- هدف از انجام این آزمایش طراحی یک کرنومتر دیجیتال با دقت یک هزارم ثانیه به کمک تایمر در میکروکنترلر STM32F401RE است که حاصل باید بر روی یک LCD کاراکتری 2*16 نمایش داده شود.

سیستم از سه بخش تشکیل شده است.

- میکروکنترلر STM32F401RE
- LCD کاراکتری 2x16 (LM016L) در پروتئوس
- سه عدد کلید push-button

نحوه کارکرد این سیستم در ادامه بیان شده است.

مدل نمایش این شمارنده دیجیتال باید به شکل زیر باشد.

MM:SS:mmm

که mmm از ۰ تا ۹۹۹ را باید با فاصله زمانی یک هزارم ثانیه (یک میلی ثانیه) بشمارد و پس از آن به مقدار SS باید ۱ واحد اضافه شود. هنگامی که SS به مقدار ۵۹ رسید یک دقیقه می شود و به MM باید ۱ واحد اضافه شود.

❖ سیستم باید بلافاصله پس از روشن شدن عبارت Welcome را بر روی نمایشگر نشان دهد. هنگامی که کلید یک فشرده شد، شمارش صعودی بر روی نمایشگر آغاز می شود. فاصله زمانی هر دو شمارش متوالی یک هزارم در نظر گرفته شود.

❖ سیستم باید قادر باشد هنگامی که شمارنده در حال شمارش است اگر کلید دو، فشرده شود. شمارنده را متوقف نماید و تا زمانی که شمارنده دستور شمارش مجدد را از کلید یک و یا ریست را از کلید سوم دریافت نکرده است آخرین مقدار شمارش شده بر روی نمایشگر باقی بماند.

❖ سیستم باید قادر باشد هنگامی که شمارنده در حال شمارش است اگر کلید سه فشرده شود شمارش متوقف شود و تا زمانی فرمان شمارش مجدد را از کلید یک دریافت نکرده است مقدار ۰۰:۰۰:۰۰۰ بر روی lcd نمایش داده شود.

❖ همچنین سیستم باید قادر باشد اگر کلید سه، سه هزار میلی ثانیه (سه ثانیه) به صورت ممتد فشرده شده است شمارش را متوقف کند و بر روی نمایشگر Turn Off را نمایش دهد. در این حالت تا زمانی که سیستم بازنشانی نشود (ریست) هیچ فرمان جدیدی را نپذیرد.

❖ عبارت Turn Off باید به صورت چشمک زن و با فاصله هر چشمک ۵۰۰ میلی ثانیه بر روی صفحه نمایش داده شود.

❖ باید ورودی هر سه کلید به صورت وقفه خارجی تشخیص داده شود. بین کلیدهای اول و دوم به لبه پایین رونده حساس هستند و بین کلید سوم هم به لبه پایین رونده و هم به لبه بالا رونده حساس است. برای کلید سوم هنگامی که یک لبه تشخیص داده شد (هنگام فشرده شدن کلید)، در سرویس روتین مربوط به آن یک تایمر فعال شود. اگر در فاصله کمتر از سه ثانیه لبه دیگر تشخیص داده شد (هنگام رها سازی کلید) آنگاه سیستم باید مطابق با فشردن معمولی عمل کند در غیر این صورت اگر بعد از سه ثانیه لبه دیگر تشخیص داده نشد (کاربر حداقل سه ثانیه کلید را فشرده نگه دارد) سیستم باید مطابق با فشردن ۳ ثانیه ای کلید سوم رفتار کند.

تذکرات مهم

- برای حفظ و تمرین ماژولاریتی، عملکردهای مختلف سیستم باید به صورت توابع مستقل و مجزا نوشته شوند.
- شماره تایمرهای مورد استفاده در گزارش کار آورده شود.
- تنها به کارگیری کتابخانه CMSIS در این آزمایش‌ها مجاز است و به کارگیری توابع HAL مجاز نیست.

موارد تحویل دادنی

- سورس کد تمام بخش‌های ذکر شده را به صورت کامل تحویل دهید. برای خوانایی بیشتر باید بخش‌های مختلف کد کامنت گذاری شود.
- پروژه ساخته شده در Proteus را باید تحویل دهید.
- گزارشی کامل و روشن از بخش‌های مختلف انجام شده در طی اجرای دستور کار تحویل شود. اگر در بخشی قطعه کدی توضیح داده می‌شود کپی آن بخش از کد در گزارش آورده شود.
- شماره پین‌ها و پورت‌های به کار گرفته شده به همراه نوع تنظیماتی که برای آن لحاظ شده است در گزارش بیان شود.

نکات مهم

- بخش‌های مختلفی که باید تحویل داده شوند همگی در یک فایل فشرده باشند و نام فایل فشرده در قالب زیر باشد.
<گروه درسی-نام-نام خانوادگی-شماره دانشجویی>
- به ازای هر روز تأخیر، روز اول ۱۵٪، روز دوم ۲۵٪ و روزهای سوم و چهارم ۳۰٪ از نمره کسر خواهد شد و در روز پنجم نمره‌ای تخصیص نمی‌گردد.
- دقت شود که در گزارش نام اعضا، شماره دانشجویی و گروه درسی ذکر گردد.
- آزمایش‌های ریزپردازنده به صورت گروه‌های دونفره انجام داده شده و تحویل می‌شوند.
- نکته مهم این است تمامی افراد گروه باید به همه جوانب و جزئیات آزمایش‌ها مسلط باشند که این نکته توسط مدرسین هنگام تحویل به دقت بررسی خواهد شد.
- هر گروه باید به صورت مجزا آزمایش را انجام دهد. کپی نتایج آزمایش گروه‌های دیگر تخطف است.
- به منظور ایجاد شرایط یکسان برای تمامی گروه‌ها و فاصله داشتن زمان آپلود و تحویل، به هنگام تحویل، اعضای گروه، در همان زمان پاسخ آزمایش خود را از درس‌افزار داندلود کرده و روی سیستم خود تحویل می‌دهند.

موفق باشید

گروه آزمایشگاه‌های ریزپردازنده