گزارش آزمایش ششم آزمایشگاه ریزپردازنده

اعضای گروه: مهدیه عربی، عرفان ربانی

سوالات تحلیلی:

سوال اول) حالت Capture Input در تایمر یکی از حالت‌های کاربردی تایمر در میکروکنترلرهاست که برای اندازه‌گیری دقیق زمان وقوع یک رویداد خارجی طراحی شده است. در این حالت، تایمر به صورت مداوم شمارش می‌کند و با وقوع یک تغییر مشخص در سیگنال ورودی مانند لبه بالا رونده یا پایین رونده روی یک پایه GPIO خاص مقدار فعلی تایمر به‌طور خودکار در یک رجیستر به نام Capture ذخیره می‌شود. این مقدار نشان‌دهنده زمان دقیق وقوع آن رویداد نسبت به شروع شمارش تایمر است. با استفاده از این قابلیت می‌توان پارامترهایی مانند دوره تناوب(Period) ، فرکانس، و پهنای پالس (Pulse Width) یک سیگنال ورودی را محاسبه کرد. برای مثال، اگر لبه‌های متوالی یک پالس را ثبت کنیم، اختلاف بین دو مقدار Capture به ما دوره تناوب سیگنال را می‌دهد. این ویژگی معمولاً در کاربردهایی مانند اندازه‌گیری سرعت چرخش موتور، تحلیل سیگنال‌های دیجیتال یا خواندن خروجی سنسورهای زمان‌محور استفاده می‌شود.

سوال دوم) تایمر Watchdog نوع خاصی از تایمر در میکروکنترلرهاست که برای افزایش پایداری، ایمنی و اطمینان از عملکرد صحیح سیستم طراحی شده است. هدف اصلی آن این است که سیستم را در برابر قفل شدن (Hanging) یا بروز خطاهای نرم‌افزاری که باعث توقف یا گیر کردن برنامه می‌شوند، محافظت کند . Watchdogبه گونه‌ای کار می‌کند که پس از فعال‌سازی، یک تایمر داخلی شروع به شمارش معکوس می‌کند و اگر در مدت زمان مشخصی توسط برنامه ریست یا رفرش نشود، فرض می‌کند که سیستم دچار اختلال شده و به صورت خودکار یک ریست کامل سیستم انجام می‌دهد. این ریست می‌تواند سیستم را به حالت اولیه بازگرداند و از باقی ماندن آن در یک وضعیت ناپایدار جلوگیری کند. برای استفاده از Watchdog، باید در ابتدا آن را کانفیگ کرد، مقدار تایم‌اوت را تنظیم نمود (مثلاً از طریق پری‌اسکیلر یا لود کردن مقدار اولیه)، و سپس در حلقه اصلی برنامه به‌صورت منظم آن را بازنشانی کرد. اگر برنامه دچار مشکل شود و نتواند این کار را انجام دهد، Watchdogبه عنوان یک مکانیسم ایمنی وارد عمل شده و سیستم را ریست می‌کند.

عملی:

متغیرهایی برای مدیریتUART ، تایمر و همچنین یک رشته برای ارسال داده از طریق UART تعریف شده‌اند. متغیر freqHzنیز برای ذخیره مقدار فرکانس اندازه‌گیری‌شده تعریف شده و به‌صورت volatile نوشته شده است تا در وقفه‌ها نیز به‌درستی قابل استفاده باشد. در تابع main ابتدا توابع اولیه‌سازی سخت‌افزار صدا زده می‌شوند که شامل تنظیم کلاک سیستم، UART، GPIOو تایمر مربوط به PWM است. پس از آن، تولید سیگنال PWM از طریق HAL_TIM_PWM_Startبرای تایمر ۳ و کانال ۱ فعال می‌شود. سیگنال تولیدشده روی پایه PA6 قرار دارد که به پایه PA1 به عنوان ورودی وقفه متصل خواهد شد. یک پیام "READY"از طریق UART ارسال می‌شود تا اطمینان حاصل شود ارتباط سریال به‌درستی برقرار است.

حلقه اصلی برنامه یک تایمر نرم‌افزاری بر پایه HAL_GetTick()پیاده‌سازی کرده است که هر ۱۰۰۰ میلی‌ثانیه یک‌بار مقدار فعلی فرکانس اندازه‌گیری‌شده را به‌صورت یک رشته متنی مانند "FREQ: 980 Hz"از طریق UART برای کامپیوتر ارسال می‌کند. این کار باعث می‌شود کاربر در هر لحظه از مقدار فرکانس سیگنال مطلع شود.

تابع HAL_GPIO_EXTI_Callbackبه‌عنوان تابع کال‌بک برای وقفه پایه PA1 تعریف شده است. این تابع هر بار که لبه بالارونده در پایه PA1 رخ دهد، فعال می‌شود. در این تابع، زمان فعلی سیستم توسط HAL_GetTick()گرفته شده و اختلاف آن با آخرین زمان ثبت‌شده در lastEdgeTickمحاسبه می‌شود. از آنجا که خروجی HAL_GetTickبه میلی‌ثانیه است، برای تبدیل دوره تناوب به فرکانس از فرمول 1000 / diffاستفاده شده است. سپس lastEdgeTickبه مقدار فعلی زمان بروزرسانی می‌شود تا در وقفه بعدی دوباره مقایسه انجام شود.

در تابع MX_TIM3_Initتنظیمات مربوط به تایمر انجام می‌شود. تایمر ۳ با پری‌اسکیلر ۷۱ و پریود ۹۹۹ پیکربندی شده است که در نتیجه فرکانس خروجی PWM برابر با حدود ۱ کیلوهرتز می‌شود. مقدار پالس در کانال PWM برابر با ۵۰۰ در نظر گرفته شده تا duty cycle برابر با ۵۰ درصد باشد. پایه PA6 برای خروجی PWM تنظیم شده و به حالت alternate function در آمده است.

تابع MX_GPIO_Initپایه PA1 را به عنوان ورودی وقفه (EXTI1) با لبه بالا تنظیم می‌کند. همچنین وقفه آن در کنترلر NVIC فعال می‌شود تا سیستم بتواند به تغییرات در این پایه واکنش نشان دهد. تنظیمات UART2 نیز در MX_USART2_UART_Initصورت گرفته و پایه‌های PA2 و PA3به‌عنوان TX و RX انتخاب شده‌اند.

تابع EXTI1_IRQHandlerنیز وقفه سخت‌افزاری EXTI1 را مدیریت می‌کند و با فراخوانی HAL_GPIO_EXTI_IRQHandlerمسیر وقفه به تابع کال‌بک هدایت می‌شود. در نهایت، تابع

Error_Handlerدر صورت بروز خطا پردازنده را در یک حلقه بی‌نهایت نگه می‌دارد و وقفه‌ها را غیرفعال می‌کند.