

بسمه تعالی

گزارش آزمایش ۹ آزمایشگاه ریزپردازنده

سیدمحمد رضا حسینی

حسین حاجی رومنان

سوال اول: روش های مختلف تبدیل آنالوگ به دیجیتال را شرح دهید.

۱. فلش ADC (نوع مستقیم): ADCهای فلش که "ADCهای مستقیم" نیز نامیده می شوند و بسیار سریع هستند و قادر به نمونه برداری در محدوده گیگاهرتز هستند. آنها با مجموعه ای از مقایسه کننده ها که به صورت موازی کار می کنند و هر کدام برای یک محدوده ولتاژ تعریف شده به این سرعت ها دست می یابند. در نتیجه، آنها در مقایسه با سایر ADCها بزرگ و گران هستند. آنها به $2^N - 1$ مقایسه کننده نیاز دارند، که در آن N تعداد بیت ها است (بنابراین برای وضوح ۸ بیت به ۲۵۵ مقایسه کننده نیاز دارد).
۲. ADC نیمه فلش: این نوع ADC ها محدودیت اندازه ADC فلش را با استفاده از دو مبدل فلش مجزا که هر کدام با وضوح نیم بیت هستند، برطرف می کنند. یک مبدل فلش بیت های با ارزش را مدیریت می کند در حالی که دیگری با بیت های کم ارزش را مدیریت میکند و برای مثال وضوح ۸ بیت با ۳۱ مقایسه کننده قابل حصول است.
۳. تقریب متوالی (SAR): این ADC ها از یک مقایسه کننده برای مقایسه ولتاژ ورودی و خروجی مبدل داخلی دیجیتال به آنالوگ استفاده می کنند و به طور متوالی قضاوت می کنند که آیا ورودی بالاتر یا پایین تر از نقطه میانی هر محدوده است. به عنوان مثال، یک سیگنال ورودی ۵ ولت بالاتر از نقطه میانی محدوده ۰ تا ۸ ولت است (نقطه میانی ۴ ولت است). بنابراین، ما سیگنال ۵ ولت را با محدوده ۴ تا ۸ ولت مقایسه می کنیم که کمتر از نقطه میانی محدوده است. این روند را ادامه می دهیم تا رزولوشن به حداکثر برسد یا به وضوح مورد نظر برسید.
۴. Sigma-Delta ADC: یک طراحی ADC نسبتاً جدید است. Sigma Deltas در مقایسه با طراحی های دیگر بسیار کند هستند اما بالاترین وضوح را در بین انواع ADC ارائه می دهند. در نتیجه، آنها برای برنامه های صوتی با کیفیت بالا بسیار مناسب هستند، اما معمولاً در جاهایی که پهنای باند بیشتری لازم است (مانند ویدیو) قابل استفاده نیستند.
۵. ADC خط لوله^۱: ADC های خط لوله که "کوانتیزرهای فرعی" نیز نامیده می شوند، از نظر کارکرد شبیه به SAR ها هستند. ابتدا یک تبدیل تقریبی کلی انجام میدهند و سپس با کوچک کردن بازه تبدیل، به تبدیل دقیق تری دست میابند. ADC های خط لوله معمولاً حد وسطی بین SAR ها و ADC های فلش هستند که سرعت متعادل، وضوح و اندازه بالا ارائه می دهند.

^۱ Pipelined

سوال دوم: مدهای کاری مختلف تبدیل آنالوگ به دیجیتال در میکروکنترلرهای STM32 را نام برده و توضیح دهید.

میکروکنترلرهای STM32 از پیشرفته ترین ADC های موجود در بازار میکروکنترلرها را دارند. میتوان کاربردهای زیادی را بر اساس ویژگی های ADC این میکروکنترلر متصور شد. به طور کلی ADC این میکروکنترلر در دو مد مستقل^۲ و مد دوگانه^۳ کار میکند. مد اول کاربردهایی هستند که فقط با یک ADC کار میکنند. مد دوم حالت هایی هستند که باید دو ADC استفاده شوند (ADC1 و ADC2 به طور مشترک کار می کنند).

حالت های مستقل: حالتی است که فقط از یک ADC استفاده می شود.

۱. Single-channel, single conversion mode: این ساده ترین حالت ADC است. در این حالت، ADC تبدیل یک نمونه از کانال x را انجام می دهد و پس از تکمیل تبدیل، متوقف می شود.
۲. Multichannel (scan), single conversion mode: این حالت برای تبدیل چند کانال به صورت متوالی در حالت مستقل استفاده می شود که میتوان به کمک ترتیب دهنده^۴ ADC، ترتیب این کانال ها را مشخص کرد.

۳. Single-channel continuous conversion mode: یک کانال را به طور مداوم و نامحدود تبدیل می کند. ویژگی این حالت به ADC اجازه می دهد تا در پس زمینه کار کند. ADC کانال ها را به طور مداوم و بدون دخالت CPU تبدیل می کند. علاوه بر این، DMA را می توان در حالت دایره ای استفاده کرد، بنابراین بار CPU را کاهش می دهد.

۴. Multichannel (scan) continuous conversion mode: حالت چند کاناله یا اسکن پیوسته می تواند برای تبدیل چند کانال به صورت متوالی با ADC در حالت مستقل استفاده شود. میتوان به کمک ترتیب دهنده^۵ ADC، ترتیب این کانال ها را مشخص کرد.

۵. Injected conversion mode: این حالت برای استفاده زمانی در نظر گرفته شده است که شروع تبدیل توسط یک رویداد خارجی یا توسط نرم افزار ایجاد شود. حالت تزریق شده نسبت به حالت معمولی اولویت دارد.

حالت های دوگانه: حالت های دوگانه در میکروکنترلرهای STM32 موجود است که دارای دو ADC باشند.

^۲ Independent mode

^۳ Dual mode

^۴ Sequencer

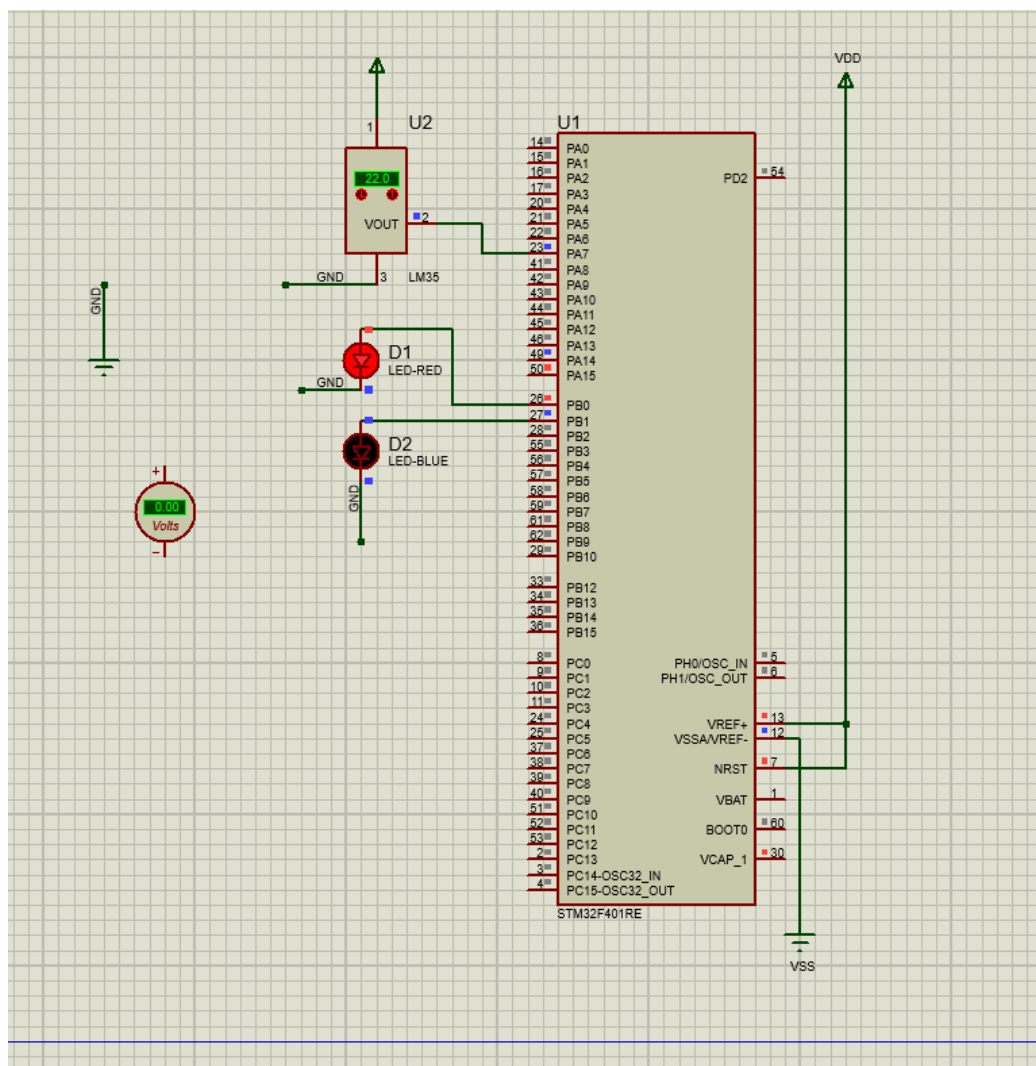
^۵ Sequencer

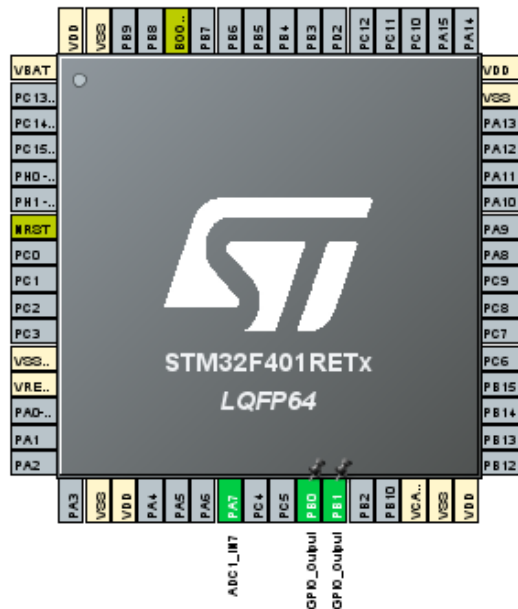
۱. Dual regular simultaneous mode: در این حالت ADC با همگام سازی ADC₁ و ADC₂ میتوان دو کانال را به صورت همزمان تبدیل کرد.
۲. Dual fast interleaved mode: حالت ADC سریع دوگانه برای تبدیل یک کانال در نظر گرفته شده است. ADC₁ و ADC₂ کانال انتخاب شده را به طور متناوب در ۷ سیکل ساعت ADC تبدیل می کنند. این بدان معنی است که کانال هر ۷ سیکل ساعت تبدیل می شود.
۳. Dual slow interleaved mode: حالت ADC آهسته دوگانه برای تبدیل یک کانال در نظر گرفته شده است. ADC₁ و ADC₂ کانال انتخاب شده را به طور متناوب در ۱۴ سیکل ساعت ADC تبدیل می کنند.
۴. Dual alternate trigger mode: در این حالت، ADC₁ و ADC₂ به طور متناوب کانال های تزریق شده را با همان تریگر خارجی مربوطه تبدیل می کنند. هنگامی که اولین تریگ رخ می دهد، تمام کانال های گروه تزریق شده در ADC₁ تبدیل می شوند. هنگامی که تریگر دوم اتفاق می افتد، تمام کانال های گروه تزریق شده در ADC₂ تبدیل می شوند.
۵. Dual combined regular/injected simultaneous mode: این حالت یک حالت همزمان منظم است که امکان تزریق را فراهم می کند. کانال های تزریق شده نیز به طور همزمان تبدیل می شوند.
۶. Dual combined: injected simultaneous + interleaved mode: این حالت ترکیبی از حالت interleaved (سریع یا آهسته) و حالت injected دوگانه همزمان است. هنگامی که کانال معمولی تریگ می شود، تبدیل دوگانه درهم شروع می شود: ADC₂ اولین تبدیل را انجام می دهد، سپس ADC₁ کانال را تبدیل می کند و به همین ترتیب ادامه پیدا میکند. هنگامی که کانال تزریق شده تریگ می شود، تبدیل کانال درهم را قطع می کند و دو ADC (ADC₁ و ADC₂) شروع به تبدیل گروه کانال های تزریق شده می کنند. هنگامی که دو ADC تبدیل تزریقی را به پایان رساندند، تبدیل به حالت interleaved بر می گردد.

سوال سوم: نرخ نایکوئیست و مفهوم آن را شرح دهید.

در پردازش سیگنال، نرخ Nyquist که به نام هری نایکوئیست نامگذاری شده است، مقداری است (بر حسب هرتز) که مقدار آن برابر است با دو برابر بالاترین فرکانس موجود در یک سیگنال معین. هنگامی که این سیگنال با نرخ بالاتری نمونه برداری شود، مجموعه دیتای بدست آمده، عاری از اعوجاجی است که به آن aliasing گویند و در صورتی که نرخ نمونه برداری کمتر از نرخ Nyquist باشد، سیگنال دارای aliasing خواهد بود. در واقع اگر سیگنالی را با فرکانسی کمتر از نرخ Nyquist نمونه برداری کنیم، نه تنها اطلاعات موجود در فرکانس بالاتر از نصف نرخ نمونه برداری را از دست می‌دهیم، بلکه همین اطلاعات دچار پدیده‌ای به نام aliasing شده و اطلاعات بدست آمده در فرکانس‌های پایین‌تر (یعنی فرکانس‌هایی که کمتر از نصف نرخ نمونه برداری هستند) را نیز خراب می‌کند. یعنی اطلاعات فرکانس‌های بالاتر خود را به عنوان اطلاعاتی با فرکانس پایین‌تر جا می‌زنند. در نهایت سیستم ما اطلاعاتی که در فرکانس بالاتر از دوبرابر نرخ نمونه برداری هستند را به اشتباه و در فرکانس اشتباهی تشخیص می‌دهد و اطلاعات نهایی را خراب می‌کند. به همین علت حتما باید سیگنال را قبل از اعمال به ADC، فیلتر کرد.

سوالات عملی





اینتراپت تایمر به اینصورت تنظیم شده که در هر یک ثانیه مقدار ورودی را از ADC بگیریم . بعد از گرفتن مقدار دیجیتال آن را با استفاده از فرمول گرفته شده به مقدار دما تغییر میدهم.

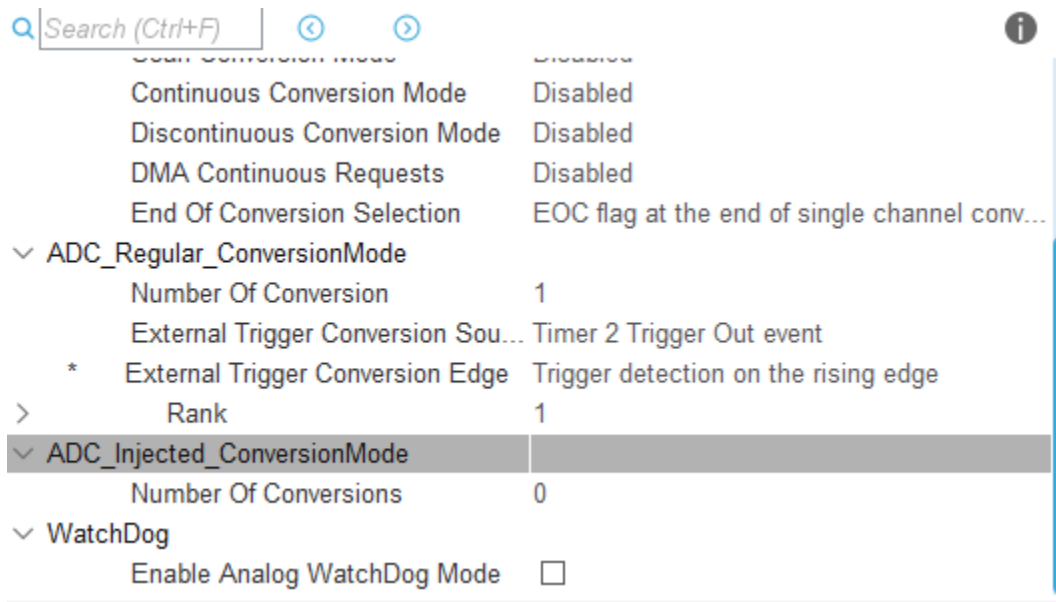
```
void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef* htim) {

    HAL_ADC_Start(&hadcl);
    HAL_ADC_PollForConversion(&hadcl, 100);
    HAL_ADC_Stop(&hadcl);
    ADC_Value = HAL_ADC_GetValue(&hadcl);
    volt = (float) (ADC_Value*3.3)/4095;
    temprature = volt*100;
    if (temprature>27){
        HAL_GPIO_WritePin(GPIOB,GPIO_PIN_2,1);
        HAL_GPIO_WritePin(GPIOB,GPIO_PIN_1,0);
    }
    else{
        HAL_GPIO_WritePin(GPIOB,GPIO_PIN_2,0);
        HAL_GPIO_WritePin(GPIOB,GPIO_PIN_1,1);
    }
}
```

قسمت اختیاری تمرین نیز پیاده سازی شده .

- Counter Settings
 - Prescaler (PSC - 16... 15999
 - Counter Mode Up
 - Counter Period (Aut... 999
 - Internal Clock Divisi... No Division
 - auto-reload preload Disable
 - Trigger Output (TRGO) Par...
 - Master/Slave Mode ... Enable (Trigger delayed for ma...
 - Trigger Event Select... Update Event

در این قسمت تایمر را جوری تنظیم میکنیم که به ازای update event ، Trigger کند و در ADC نیز تنظیمات را جوری انجام میدهیم که به ازای timer ۲ trigger out شروع به کار کند .



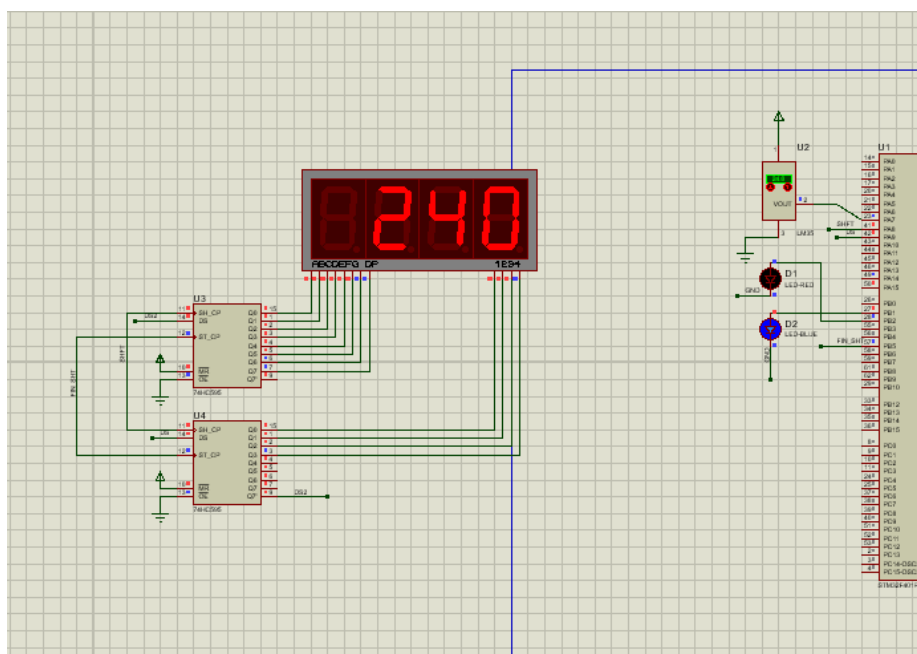
با فعالسازی اینتراپت ADC پس از تبدیل کردن نرخ آنالوگ ورودی ، میتوان مقدار دما را در IRQ_Handler ، ADC خواند.


```

*/
void HAL_ADC_ConvCpltCallback(ADC_HandleTypeDef* hadc)
{
    // Read & Update The ADC Result
    ADC_RES = HAL_ADC_GetValue(&hadc1);
    volt = (float) (ADC_RES*3.3)/4095;
    temprature = volt*100;
    if (temprature<27){
        HAL_GPIO_WritePin(GPIOB,GPIO_PIN_0,1);
        HAL_GPIO_WritePin(GPIOB,GPIO_PIN_1,0);
    }
    else{
        HAL_GPIO_WritePin(GPIOB,GPIO_PIN_0,0);
        HAL_GPIO_WritePin(GPIOB,GPIO_PIN_1,1);
    }
}

```

(۲)



مطابق بخش اول ابتدا دما را از روی سنسور LM35 میخوانیم و بر روی ۷ سگمنت ها به نمایش میگذاریم. برای استفاده از این نوع ۷segment ها ، از ۲ ، ۷۴۵۹۵ استفاده شده است . نوع کار این شیفت رجیستر ها بدین صورت است که یک بیت سریال ورودی میگیرند و با فعال شدن پین شیفت مقدار ورودی دیتا را یک واحد شیفت میدهند. پس از اینکه هشت دیتا بدین صورت ذخیره شد ، با فعال کردن پین STCP مقادیر بر روی پروت های خروجی قرار میگیرند.

با توجه به اینکه برد digital tube از پین های A۹ برای DS ، A۸ برای CLK و B۵ برای LATCH استفاده میکند و برد STM۳۲F۴۰۱RE ، SPI بر روی پین A۹ ندارد خودمان یک تابع برای دادن مقدار به صورت سریال پیاده سازی کردیم .

در این تابع مقداری که میخواهیم نمایش دهیم به عنوان ورودی اول تابع و شماره سون سگمنت به عنوان ورودی دوم داده میشود .

```

void write_serial(int input,int num){
    HAL_GPIO_WritePin(GPIOB,LATCH_Pin,0);
    int j = input;
    for(int i=0;i<8;i++){
        int shift = j%2;
        j=j>>1;
        HAL_GPIO_WritePin(GPIOA,CLK_Pin,0);
        if (shift == 1 ){
            HAL_GPIO_WritePin(GPIOA,DATA_Pin,1);
        }
        if (shift == 0 ){
            HAL_GPIO_WritePin(GPIOA,DATA_Pin,0);
        }
        HAL_GPIO_WritePin(GPIOA,CLK_Pin,1);
        HAL_Delay(0);
    }
    j=num;
    for(int i=0;i<8;i++){
        int shift = j%2;
        j=j>>1;
        HAL_GPIO_WritePin(GPIOA,CLK_Pin,0);
        if (shift == 1 ){
            HAL_GPIO_WritePin(GPIOA,DATA_Pin,1);
        }
        if (shift == 0 ){
            HAL_GPIO_WritePin(GPIOA,DATA_Pin,0);
        }
        HAL_GPIO_WritePin(GPIOA,CLK_Pin,1);
        HAL_Delay(0);
    }
    HAL_GPIO_WritePin(GPIOB,LATCH_Pin,1);
    HAL_Delay(1);
    HAL_GPIO_WritePin(GPIOB,LATCH_Pin,0);
}

```