

به نام خدا

گزارش کار آزمایشگاه ریزپردازنده

آزمایش ۷

مدرس: مهندس بی طالبی

تارا برقیان

مهرشاد سعادت‌نی‌نیا

نیم سال اول ۱۴۰۰-۱



فهرست

۳	مقدمه :
۳	پاسخ سوالات تحلیلی:
۴	بخش کدی:
۵	چالش ها :

مقدمه :

در این آزمایش هدف آشنایی با مفاهیم و دستوارت مربوط به ADC بود.

پاسخ سوالات تحلیلی:

- هنگام کار با اکثر سنسورها خواهید دید حتی در حالتی که شما در ورودی مدار تغییر خاصی را اعمال نمی کنید مقدار ADC همواره در حال نوسان است و نویز دارد. مثلا برای سنسور IR حتی اگر جسم را ثابت نیز نگه دارید اندکی تغییر در نور محیط باعث بروز اعداد متفاوت در خروجی ADC می شود. یا در همین تمرین وقتی وقفه خارجی اعمال میشود نویز داریم. یکی از روش ها برای از بین بردن این نوع خطاها میانگین گیری است.
- در بحث پردازش سیگنال، در برخی موارد جهت افزایش سرعت پردازش و کاهش زمان پاسخدهی سیستم، نیاز است تا فرکانس نمونه برداری کاهش یابد تا سرعت بیشتر شود.

بخش کدی:

این تمرین بر خلاف سادگی چالش های بسیار زیادی برای ما به همراه داشت.

در ابتدا به بررسی دستورات مورد نیاز میپردازیم :

```

132 while (1)
133 {
134     if(rank == 1)
135     {
136         ADC_Select_CH2();
137     }
138     else
139     {
140         ADC_Select_CH5();
141     }
142 }
143
144 HAL_ADC_Start(&hadcl);
145
146 if (HAL_ADC_PollForConversion(&hadcl, 10) == HAL_OK)
147 {
148     c = HAL_ADC_GetValue(&hadcl);
149     c = c*5 / 4097;
150     c=c*1000;
151     a = (int)c;
152     if(inverse == 1)
153         c=0-c;
154
155     GPIOC->ODR |= (int) (c);
156 }
157 HAL_ADC_Stop(&hadcl);
158 GPIOC->ODR &= 0;
159 }
160
161

```

این دوتابع برای تعویض کانال بین ۲ و ۵ هستند.

شروع نمونه برداری بعد از هر بار stop

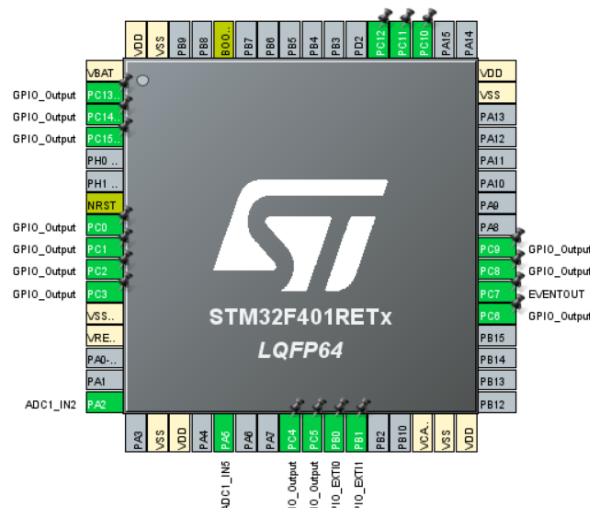
اگر time out ندهد هر ۱۰ میلی ثانیه یک بار یک نمونه برداری انجام می شود.

هر بار با تابع مورد نظر یک عدد اعشاری دریافت میشود. برای به دست آوردن مقدار درست با کم ترین خطا از فرمول زیر استفاده شده :

$$(ADC_value * vref+) / 2^{resolution}$$

بعد هر نمونه برداری باید stop و مجدد راه اندازی شود.

تنظیمات اولیه کیوب :

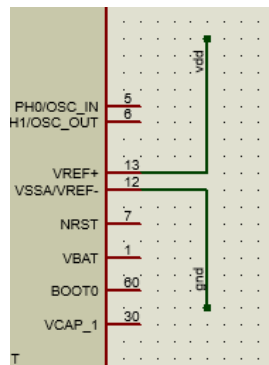


برای تشخیص فشردن کلید ها نیز از وقفه خارجی استفاده شده است.

```
void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
{
    if(HAL_GPIO_ReadPin(GPIOB, GPIO_PIN_0) == GPIO_PIN_SET) //button 1 is pressed
    {
        if(rank == 1)
            rank = 2;
        else
            rank = 1;
        //HAL_Delay(10); //debounce time
    }
    if(HAL_GPIO_ReadPin(GPIOB, GPIO_PIN_1) == GPIO_PIN_SET) //button 2 is pressed
    {
        if(inverse == 1)
            inverse = 2;
        else
            inverse = 1;
    }
}
```

چالش ها :

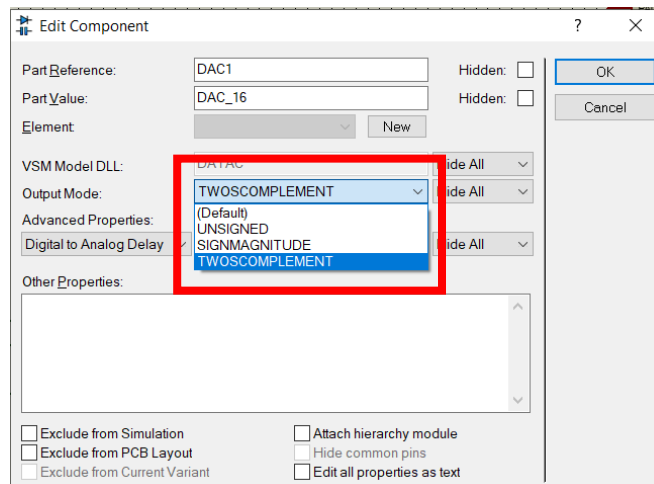
چالش های زیر نکات مهمی بودند که برای رفع هرکدام چندین ساعت زمان صرف شده و حالت های مختلف تست شده است و در اینترنت راه حل های زیر یافت نمیشود.



۱. تابع `adc_getvalu()` همواره صفر برمیگرداند که باید برای رفع آن مدار را به شکل زیر دی می آوردیم.

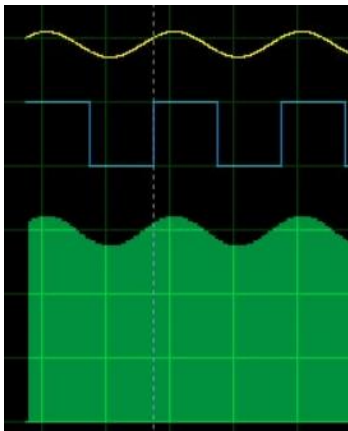
۲. ماژول `dac` برای نمایش اعداد منفی مشکل داشت که

برای رفع آن میتوان از پروتئوس سیستم نمایش اعداد آنرا تغییر داد.

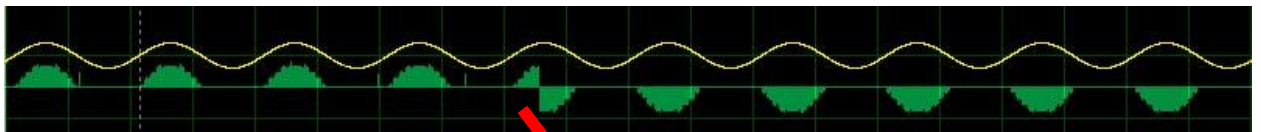


۳. اگر adc به درستی نمونه بر نمیدارد و نویز زیادی دارد احتمالا به خاطر وجود delay و یا استفاده از روش polling برای تشخیص دکمه هاست.

۴. Adc در تشخیص اعداد منفی مثلا موج سینوسی همچنان مشکل دارد ولی در نرم افزار cube برای تغییر سیستم اعداد گزینه ای نبود. البته میتوان با تغییراتی در کد به شکل زیر رسید اما باعث ایجاد نویز های بعدی و خرابی کد میشود.



در نهایت خروجی های حاصل عبارتند از : رنگ سبز حاصل نمونه برداریست.



فشردن کلید ۲



