



پروژه درس

ریزپردازنده و زبان اسمبلی

طاها موسوی 98243058

نیلوفر مرادی جم 97243063

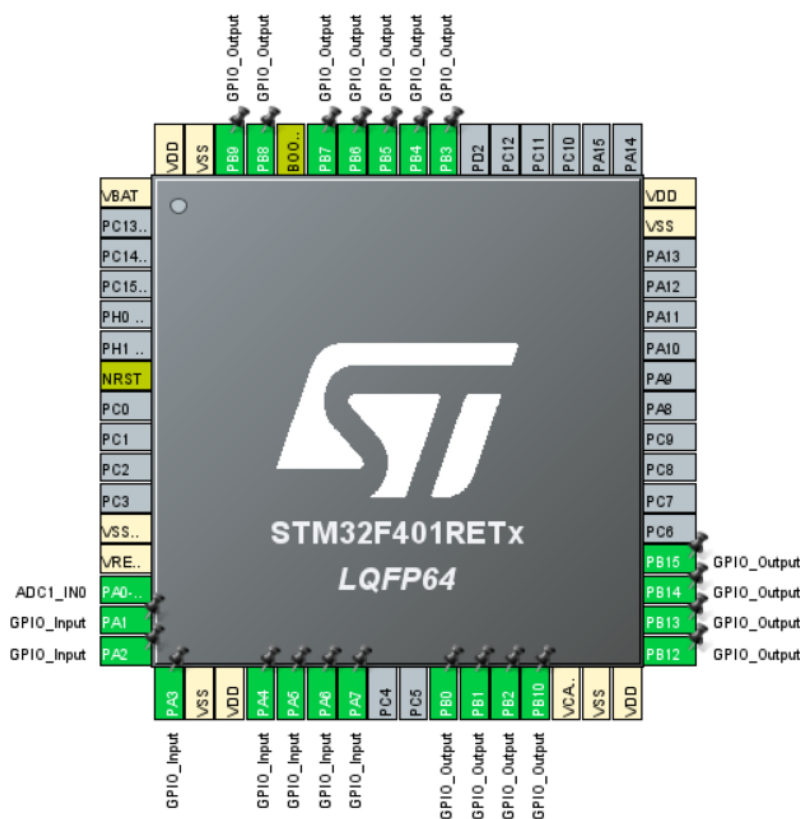
گروه 2

مقدمه

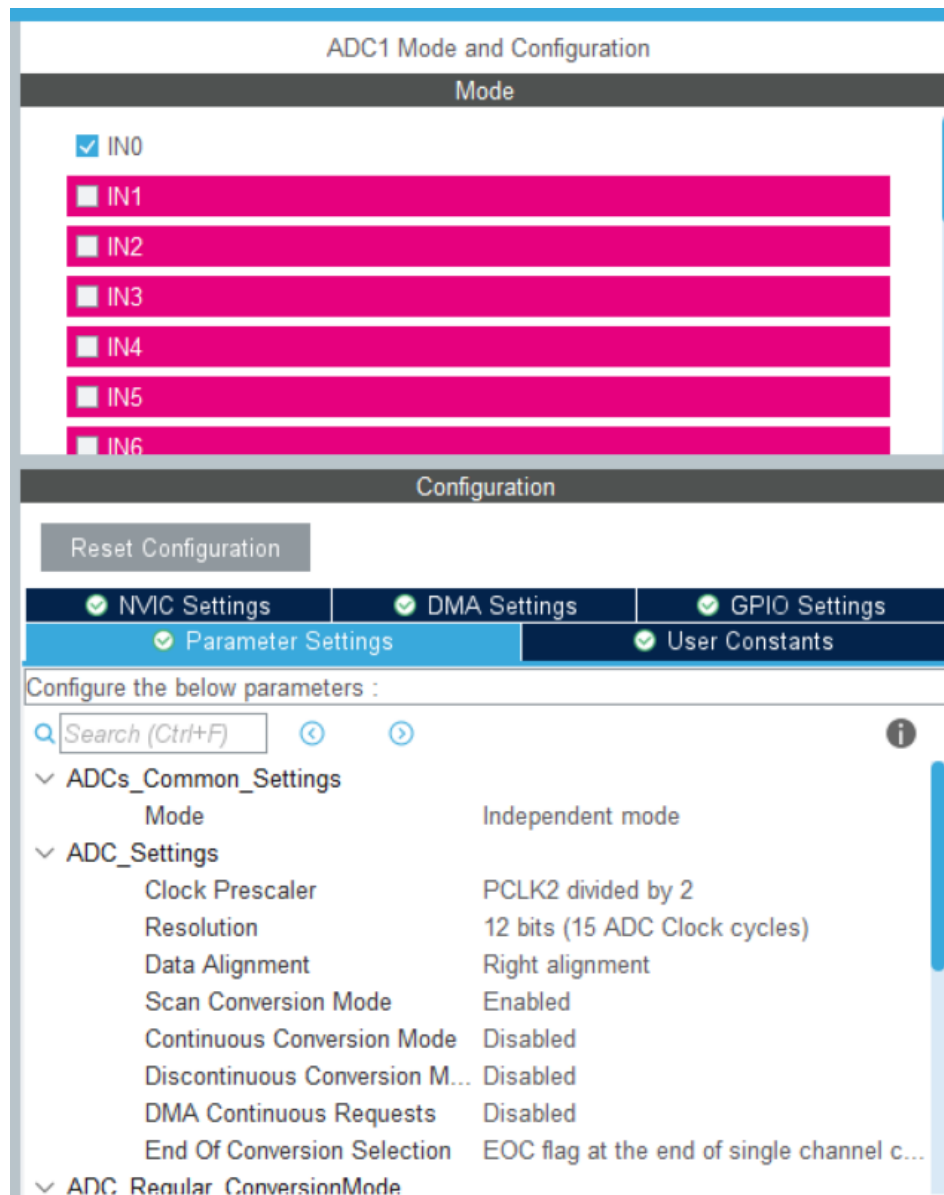
در این پروژه قصد داریم با مبدل آنالوگ به دیجیتال، سیگنال DMTF را دیکود کرده و مقدار آن را روی سون سگمنت نمایش دهیم. سیگنال های DMTF از or کردن سیگنال های سینوسی به دست می آیند.

تنظیمات cubemx

طرح کلی میکرو:



از gpio output ها (ینی pb0 تا pb6 و pb7 تا pb15) برای وصل شدن به سون سگمنت و نمایش نتیجه بر روی آن ها استفاده شده است.



در اینجا adc0 را فعال کرده ایم که به pa0 وصل می‌باشد.

جواب را در 12 بیت می‌ریزیم.

Scan conversion را enable می‌کنیم.

سمپلینگ تایم برابر 3 کلاک می‌باشد. به این معنی که 3 کلاک طول میکشد تا نمونه برداری انجام شود.

Number of conversions را برابر 1 گذاشتیم یعنی تنها یک چنل از اینجا انتخاب کردیم.

اجزا و تنظیمات پروتئوس

از اجزای زیر برای شبیه سازی در پروتئوس استفاده کردیم:

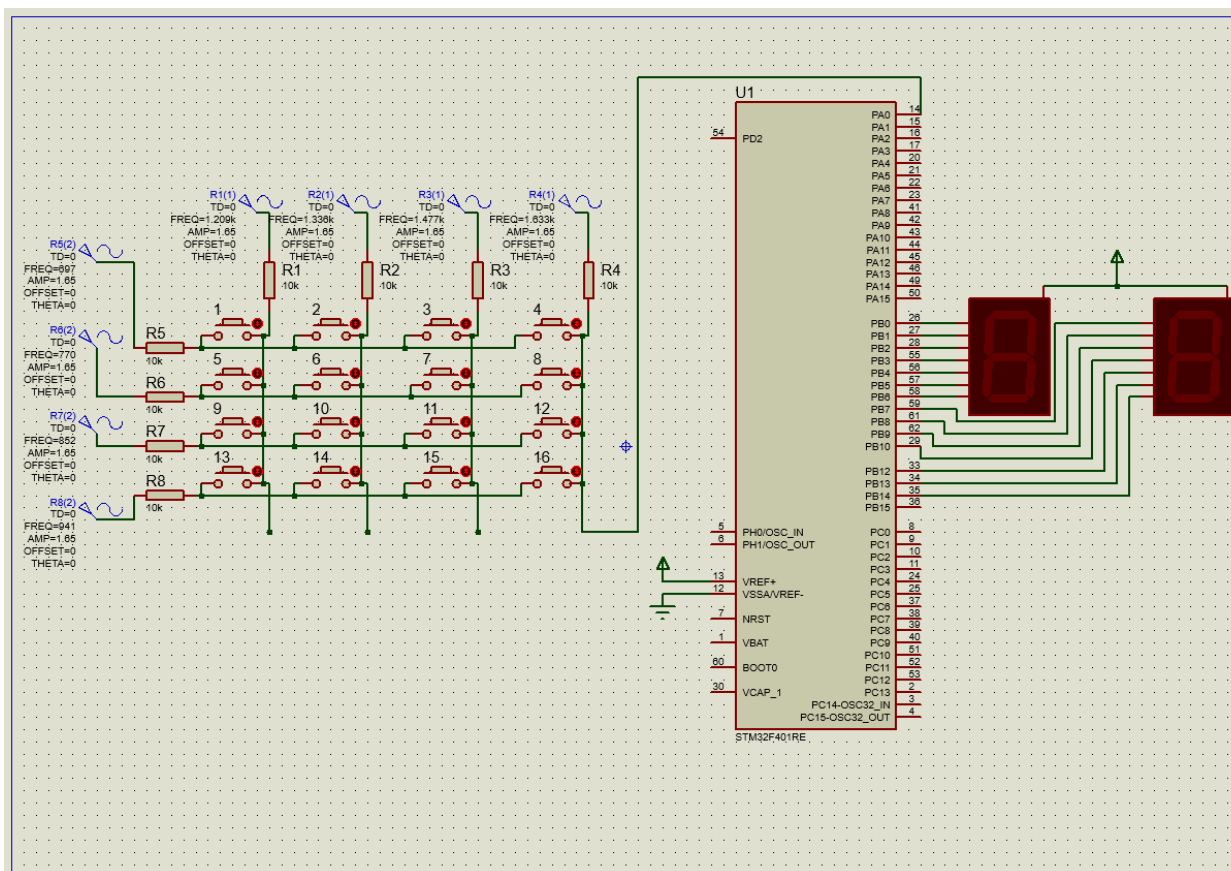
- Stm32f401re microcontroller

- مقاومت

- دو عدد 7-segment دوتایی

- ژنراتورهای تولید سیگنال

- 16 عدد کلید



همانطور که مشاهده میشود 8 تا ژنراتور تولید سیگنال هر کدام به یک مقاومت وصل شده اند و هر کلید دو تا دو تا آن ها را با هم OR میکند.

توضیحات مربوط به کد و راه حل

برای محاسبه سیگنال از الگوریتم goertzel استفاده کردیم که طبق موارد بیان شده در صورت پروژه عمل میکند.

$$\begin{aligned}Q_0 &= (coef_k \times Q_1[n]) - Q_2[n] + x[n], \\Q_1 &= Q_0[n - 1], \\Q_2 &= Q_1[n - 1], \\x[n] &= \text{نمونه خوانده شده از مبدل آنالوگ به دیجیتال} \\k &= \lceil 0.5 + ((N \times \text{فرکانس مورد جستجو}) / \text{نرخ نمونه برداری}) \rceil, \quad k \in \mathbb{N} \\N &= \text{تعداد نمونه های مجموعه مورد بررسی} \\coef_k &= 2\cos((2\pi k) / N)\end{aligned}$$

در معادلات فوق با فرض این که آرایه x به طول N نمونه از سیگنال ورودی نمونه برداری شده باشد، $x[n]$ ورودی فعلی، Q_0 خروجی فعلی و Q_1 و Q_2 خروجی های محاسبات مرحله قبل هستند. با توجه به مقادیر فوق، توان شکل موج ورودی برای فرکانس مورد جستجو در هر نوبت محاسبه عبارتست از

$$^2 \text{ دامنه} = Q_1^2[N] + Q_2^2[N] - (coef_k \times Q_1[N] \times Q_2[N])$$

```
int    k,i;
float  floatnumSamples;
float  omega,sine,cosine,coeff,q0,q1,q2,magnitude,real,imag;

float  scalingFactor = numSamples / 2.0;

floatnumSamples = (float) numSamples;
k = (int) (0.5 + ((floatnumSamples * TARGET_FREQUENCY) / SAMPLING_RATE));
omega = (2.0 * 3.1415 * k) / floatnumSamples;
sine = sin(omega);
cosine = cos(omega);
coeff = 2.0 * cosine;
q0=0;
q1=0;
q2=0;

for(i=0; i<numSamples; i++)
{
    q2 = q1;
    q1 = q0;
    q0 = coeff * q1 - q2 + adc[i];
}

// calculate the real and imaginary results
// scaling appropriately
real = (q0 - q1 * cosine) / scalingFactor;
imag = (-q1 * sine) / scalingFactor;

magnitude = sqrtf(real*real + imag*imag);
return magnitude;
};
```

منبع:

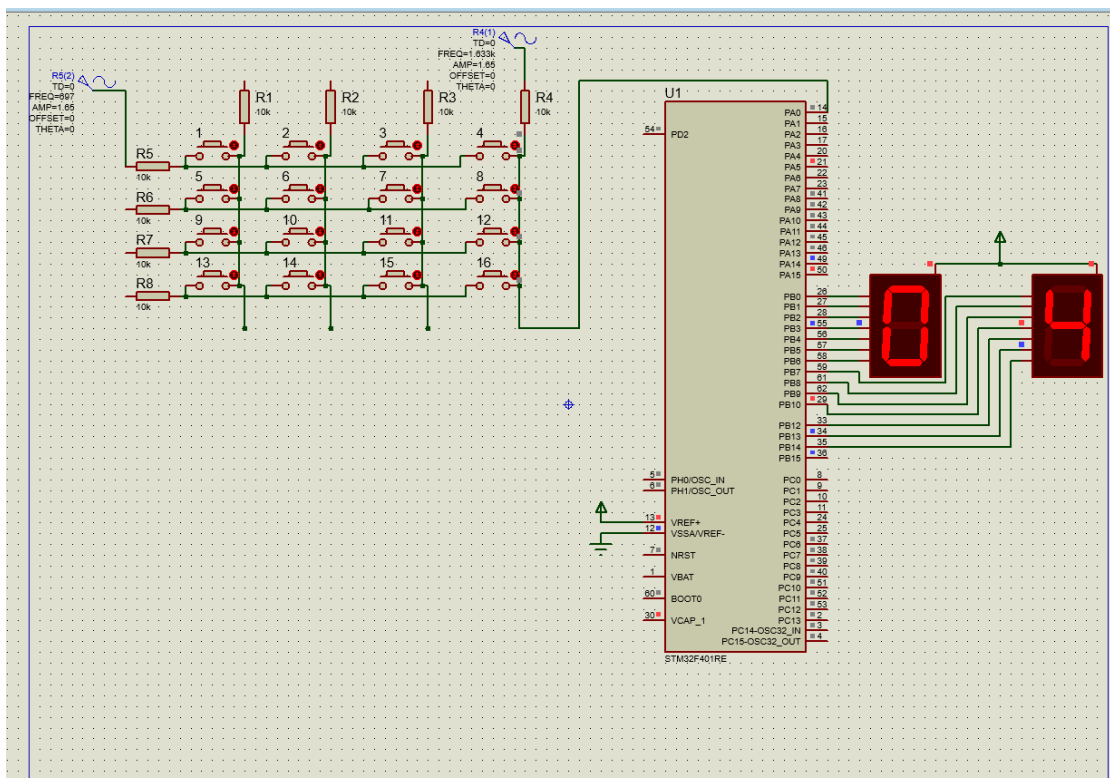
```

int main(void){
    HAL_Init();
    SystemClock_Config(); /* Configure the system clock */
    MX_GPIO_Init();
    MX_ADC1_Init();
    showOn7Segment(0xFFFF);
    hadcl.Init.NbrOfConversion=1;
    HAL_ADC_Init(&hadcl);
int numberOfShift = 7;
    while (1){
        for(i=0;i<1400;i++){
            HAL_ADC_Start(&hadcl);
            if(HAL_ADC_PollForConversion(&hadcl,1)==HAL_OK)
                adc[i]=HAL_ADC_GetValue(&hadcl);
            HAL_ADC_Stop(&hadcl);
        };
        i=0;
        if((goertzel_mag(1400,697,25000))>80.0){
            if((goertzel_mag(1400,1209,25000))>80.0)
                showOn7Segment((seg[1/10])+((seg[1]<< numberOfShift)));
            if((goertzel_mag(1400,1336,25000))>80.0)
                showOn7Segment((seg[2/10])+((seg[2]<< numberOfShift)));
            if((goertzel_mag(1400,1477,25000))>80.0)
                showOn7Segment((seg[3/10])+((seg[3]<< numberOfShift)));
            if((goertzel_mag(1400,1633,25000))>80.0)
                showOn7Segment((seg[4/10])+((seg[4]<< numberOfShift)));
        };
        i=i+1;
        if((goertzel_mag(1400,770,25000))>80.0){
            if((goertzel_mag(1400,1209,25000))>80.0)
                showOn7Segment((seg[5/10])+((seg[5]<< numberOfShift)));
            if((goertzel_mag(1400,1336,25000))>80.0)
                showOn7Segment((seg[6/10])+((seg[6]<< numberOfShift)));
            if((goertzel_mag(1400,1477,25000))>80.0)
                showOn7Segment((seg[7/10])+((seg[7]<< numberOfShift)));
            if((goertzel_mag(1400,1633,25000))>80.0)
                showOn7Segment((seg[8/10])+((seg[8]<< numberOfShift)));
        };
        i=0;
        if((goertzel_mag(1400,852,25000))>80.0){
            if((goertzel_mag(1400,1209,25000))>80.0)
                showOn7Segment((seg[9/10])+((seg[9]<< numberOfShift)));
            if((goertzel_mag(1400,1336,25000))>80.0)
                showOn7Segment((seg[10/10])+((seg[0]<< numberOfShift)));
            if((goertzel_mag(1400,1477,25000))>80.0)
                showOn7Segment((seg[11/10])+((seg[1]<< numberOfShift)));
            if((goertzel_mag(1400,1633,25000))>80.0)
                showOn7Segment((seg[12/10])+((seg[2]<< numberOfShift)));
        };
        i=0;
        if((goertzel_mag(1400,941,25000))>80.0){
            if((goertzel_mag(1400,1209,25000))>80.0)
                showOn7Segment((seg[13/10])+((seg[3]<< numberOfShift)));
            if((goertzel_mag(1400,1336,25000))>80.0)
                showOn7Segment((seg[14/10])+((seg[4]<< numberOfShift)));
            if((goertzel_mag(1400,1477,25000))>80.0)
                showOn7Segment((seg[15/10])+((seg[5]<< numberOfShift)));
            if((goertzel_mag(1400,1633,25000))>80.0)
                showOn7Segment((seg[16/10])+((seg[6]<< numberOfShift)));
        };
    };
};

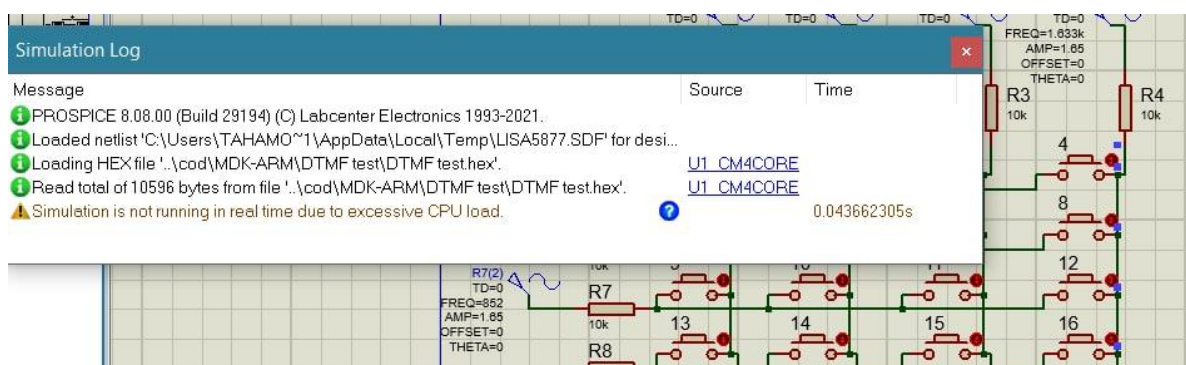
```

در تابع مین با بررسی سطر و ستون متوجه میشویم که کدام کلید فشرده شده و نتیجه مربوط به آن را با تابع showOn7Segment روی سون سگمنت نمایش می‌دهیم.

مشاهده خروجی پروژه در پروتئوس

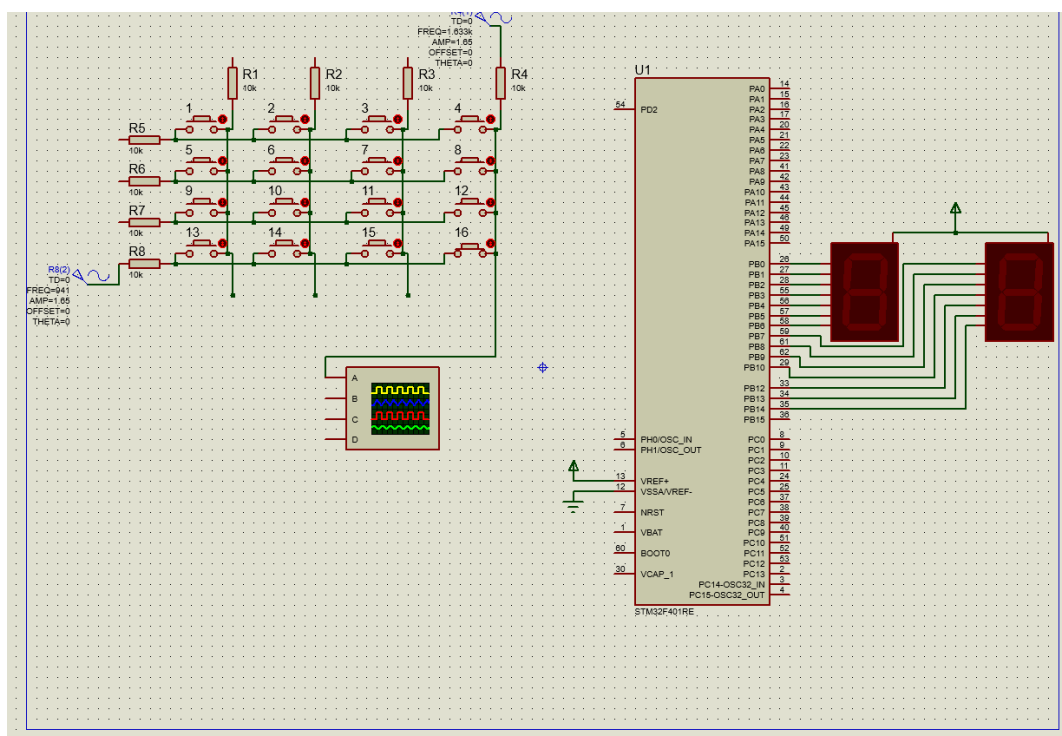


با کلیک بر روی هر دکمه، عدد معادل آن دکمه باید روی سون سگمنت نمایش داده شود. اما اینجا با یک چالش روبرو شدیم.

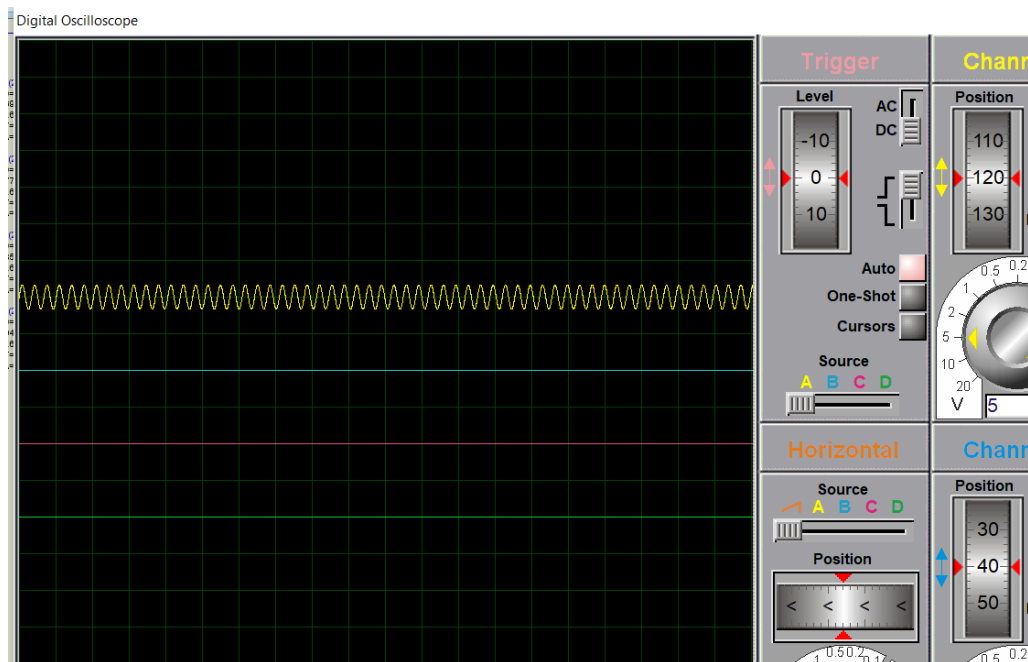


همانطور که مشاهده میشود، شبیه سازی بصورت real time اجرا نمیشود. به همین صورت سرعت شبیه سازی پس از کلیک بر روی دکمه و OR شدن سیگنال ها به شدت پایین می آید. به طوری که ممکن است چندین دقیقه طول بکشد تا نتیجه OR شدن سیگنال ها به دست آید و عدد روی سون سگمنت نمایش داده شود.

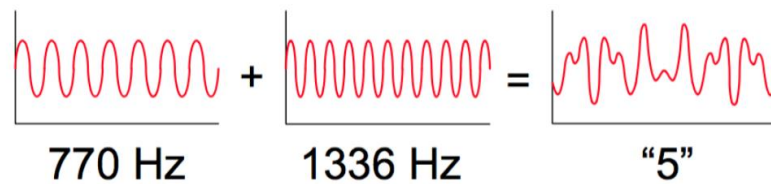
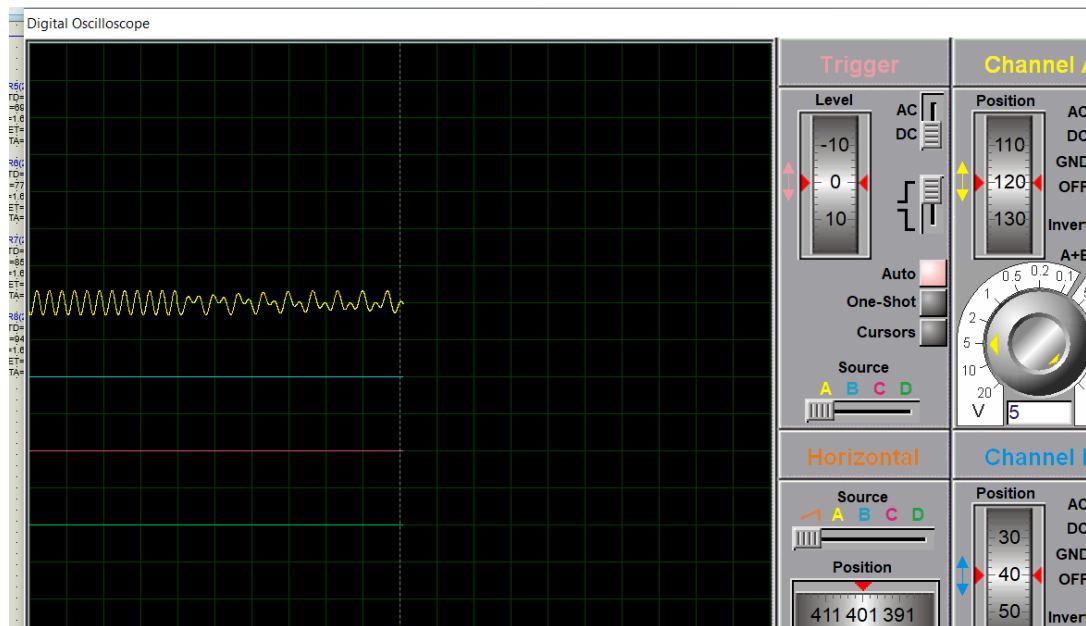
برای اینکه در این زمان مطمئن شویم که فرایند متوقف شده و انجام نمیشود یا به درستی در حال انجام هست از یک اسیلوسکوپ برای دیدن نتایج سیگنال ها استفاده کردیم.



قبل از کلیک دکمه:



بعد از کلیک دکمه:



بخاطر این چالش گرفتن خروجی ها و نمایش آن ها به شدت برای ما سخت شد. موفق شدیم چند تا خروجی رو پشت سر هم بگیریم که بصورت ویدیو در فایل پروژه پیوست شده است.