

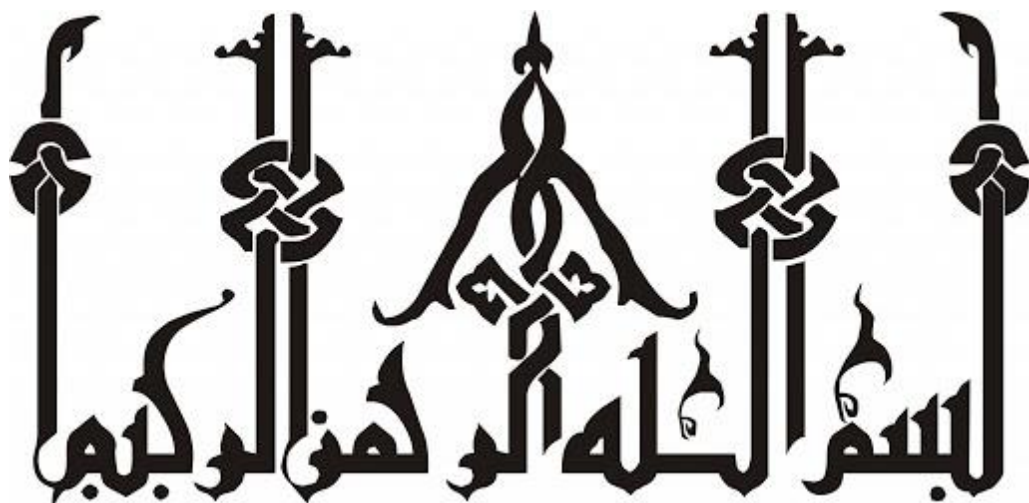


پروژه کنترل خطی

دکتر عباداللهی

آروین ستاری

تیر ۱۴۰۰



## تنظیم دور موتور DC بر اساس شدت صوت با استفاده از میکروکنترلر **Arduino Nano**

نگاهی اجمالی بر کارکرد کل مدار :

مدار ما از سه قسمت ورودی (شدت صدا)، واحد پردازش (میکروکنترلر **Arduino Nano**) و خروجی (تنظیم سرعت موتور DC) تشکیل شده است. الگوریتم کلی به اینگونه است که سرعت موتور DC متناسب با شدت صدای ورودی، افزایش و کاهش می یابد. حال به ترتیب و مرحله به مرحله هر یک از این قسمت ها را تشریح می کنیم.

**ورودی صدا :**

جهت ورودی صدا از مازول پری آمپلیفایر با تقویت کننده عملیاتی **LM386**، و با **Gain** قابل کنترل با پتانسیومتر **Multi-Turn** و مجهز به میکروفون خازنی استفاده می شود.



### ماژول میکروفون با پری آمپلیفایر با Gain متغیر

این ماژول دارای ۴ پین می باشد که به ترتیب عبارتند از:

- AO: خروجی آنالوگ صدا
- Ground: G (قطب منفی تغذیه)
- VCC: + (قطب مثبت تغذیه)
- DO: که وجود یا عدم وجود صدا را به ترتیب با سطوح دیجیتال ۱ و ۰ گزارش می کند. این پین در پروژه ما استفاده نمی شود.

بنابراین از پین AO (خروجی آنالوگ صدا) به عنوان ورودی صدا در پروژه استفاده می کنیم.

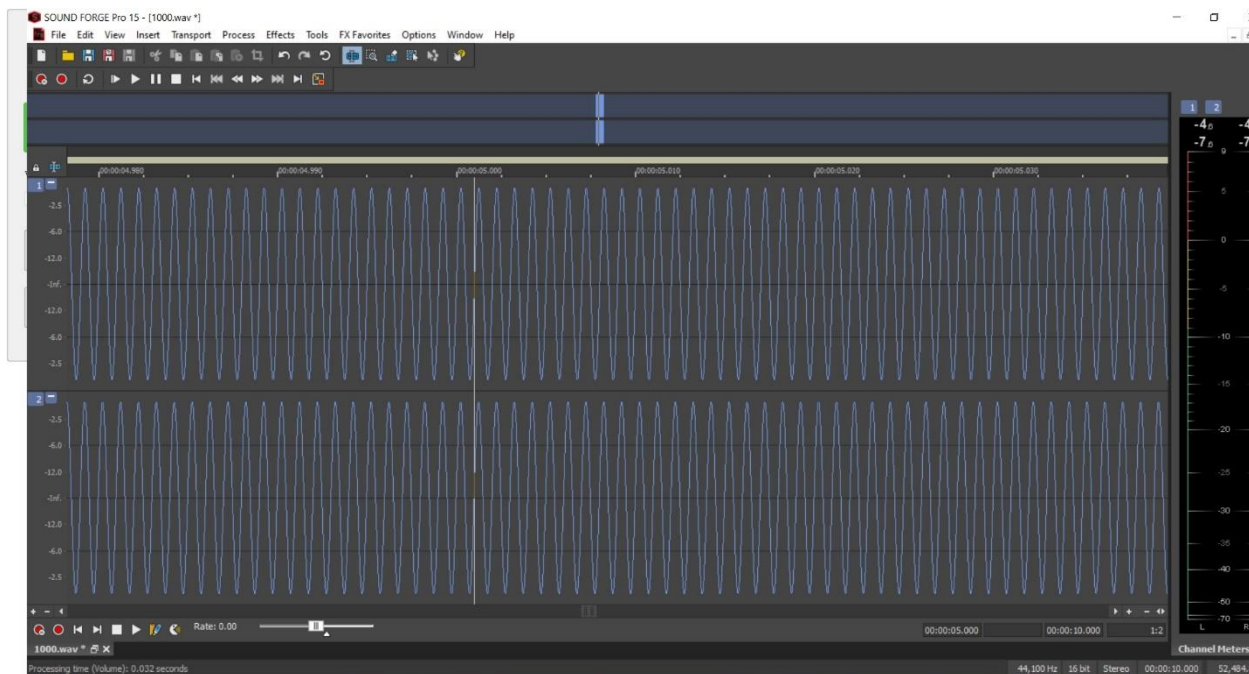
مسئله مهم دیگر این است که نرم افزار Proteus فاقد المان شبیه سازی میکروفون می باشد و تنها راه ممکن برای استفاده از سیگنال صوتی استفاده از Audio Generator می باشد که به عنوان ورودی صدا نیاز به یک فایل صوتی با پسوند WAV دارد. جهت تولید فایل صوتی مد نظر از وبسایت [online Tone Generator](#) استفاده شد.

بدین منظور با استفاده از این وبسایت، فایل صوتی WAV شامل موج سینوسی با فرکانس ۱۰۰۰ و مدت ۱۰ ثانیه، ایجاد شد.



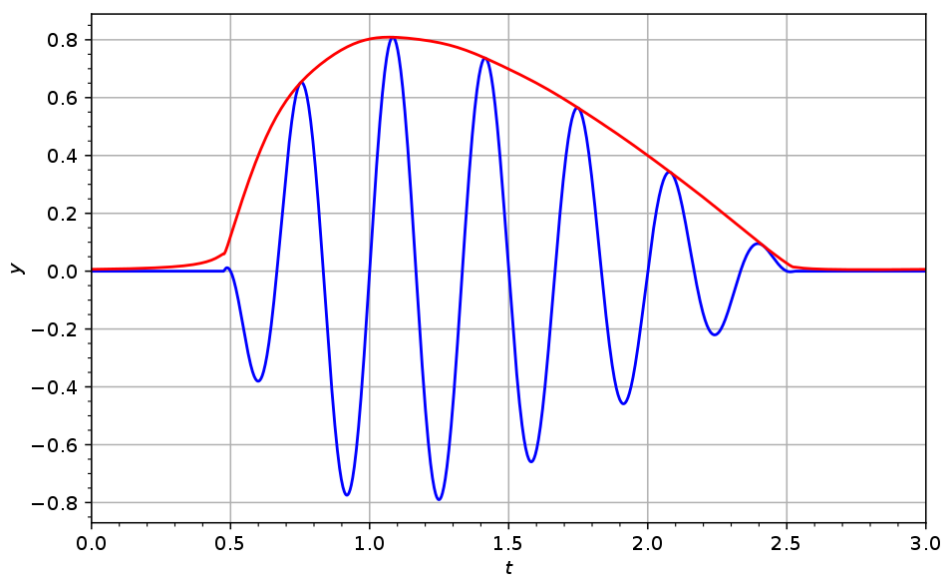
Online Tone Generator

حال نیاز داریم که دامنه (Amplitude) این موج را در ۵ ثانیه اول به صورت صعودی (Fade In) و ۵ ثانیه دوم را به صورت نزولی (Fade Out) تغییر دهیم. بدین منظور از نرم افزار ویرایش فایل های صوتی SoundForge استفاده شد.



نمای ابتدایی فایل خام با فرکانس ۱۰۰۰ هرتز (دامنه همه جا یکسان است)

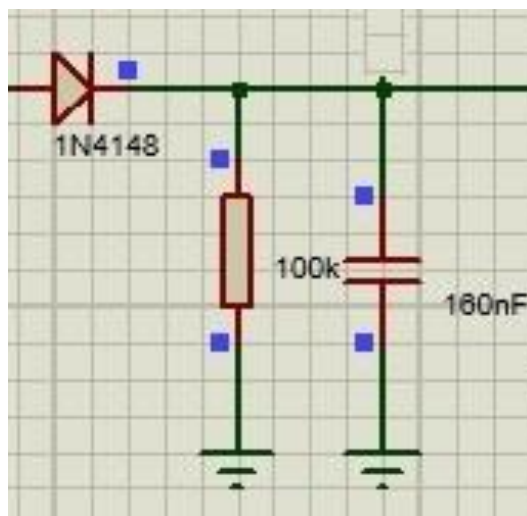
همان‌طور که مشاهده می‌شود بنا به ویژگی موج‌های سینوسی **Amplitude** سیگنال مرتب در حال تغییر می‌باشد. راهکار مناسب برای رفع این مشکل استفاده از **Envelope Detection Circuit** استفاده می‌شود.



Envelope Detection Circuit

بدین ترتیب سیگنال خروجی متناسب با Amplitude سیگنال صوتی خواهد بود که می توان به طور مستقیم به ورودی آنالوگ آردوینو متصل کرد.

طراحی مدار Envelope Detection Circuit به شکل زیر است.

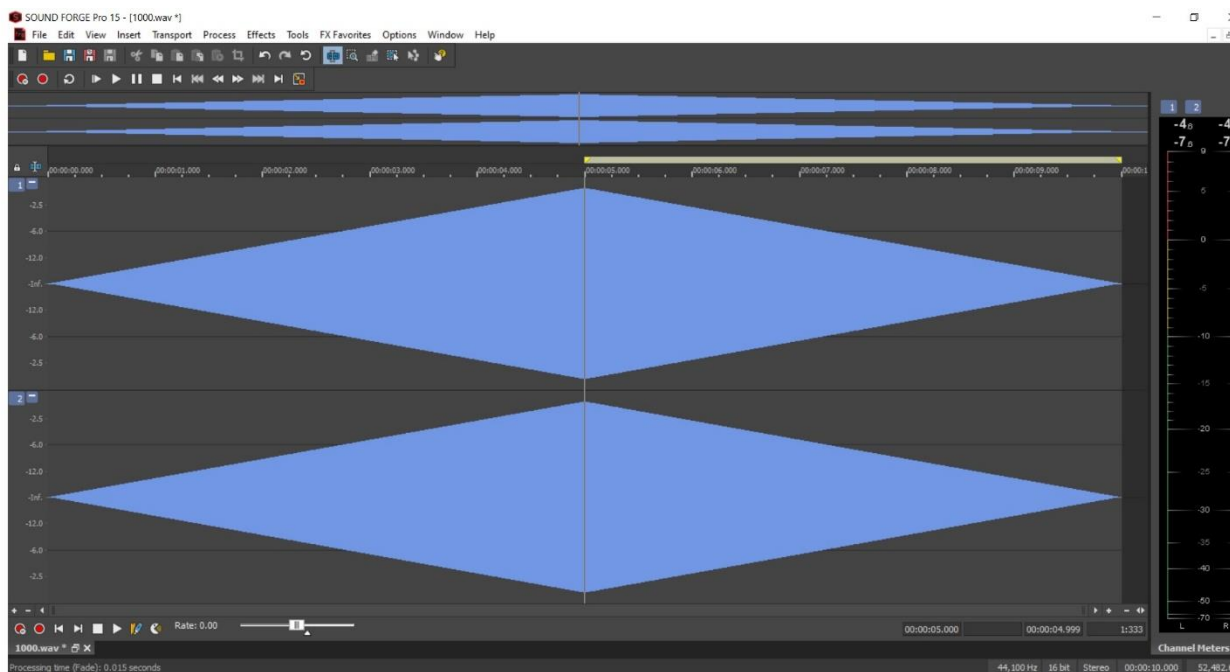


Envelope detection Circuit

قانون سرانگشتی برای انتخاب خازن و مقاومت این است که خازن و مقاومت را ابتدا بر اساس  $F_{mean}$  بر اساس فرمول زیر محاسبه می کنیم:

$$f = \frac{\sqrt{2}}{2000}$$

و  $R$  و  $C$  را ۱۰ برابر مقادیر محاسبه شده به کار می بریم.



پس از اعمال FadeIn در نیمه ابتدایی و FadeOut در نیمه دوم

بدین ترتیب به فایل صوتی با پسوند WAV با دامنه متغیر مورد نیاز برای شبیه سازی Proteus دست یافتیم.



### واحد پردازش (میکروکنترلر Arduino Nano)

این میکروکنترلر مانند یک کامپیوتر کوچک عمل می کند که با استفاده و اجرای نرم افزار C که در محیط برنامه نویسی Arduino IDE نوشته شده است، پین های ورودی و خروجی دیجیتال و ورودی های آنالوگ را کنترل می نماید.

```
int soundPin = A0;
int motorPin = 6;

int soundLevel = 0;
int motorLevel = 0;

char buff[20];

void setup() {

    // Define Pin Directions
    pinMode(motorPin, OUTPUT);

    // Initiate Output Pins
    digitalWrite(motorPin, LOW);

    Serial.begin(9600);
}

void loop() {

    /*
    * Arduino has ADC 10 bit resolution --> 0..5V translates to 0..1023 ( $2^{10}$ 
    - 1) --> Each step is ~ 5mV
    * Our Audio pre-amplifier module output is 0..2V range.
    * Our envelope detection circuit has a diode in forward bias (0.6V) in its
    input --> Envelope detection output is in 0 .. 1.4V which translates to 0..280
    * So soundLevel variable will be in 0..280 range
    */
    soundLevel = analogRead(soundPin);

    /*
    * Arduino uses PWM to simulate analog output
    * PWM signal is a square wave with a fixed frequency (490Hz in Arduino)
    and a changing DUTY CYCLE (onTime / offTime)
    * A 1 byte variable (0..255) translates to 0..100% duty cycle
    */
}
```



\*/

```
// We need to map 0..80 soundLevel to 0..255 motorLevel
motorLevel = soundLevel * 255 / 280;
analogWrite(motorPin, motorLevel & 0xFF);

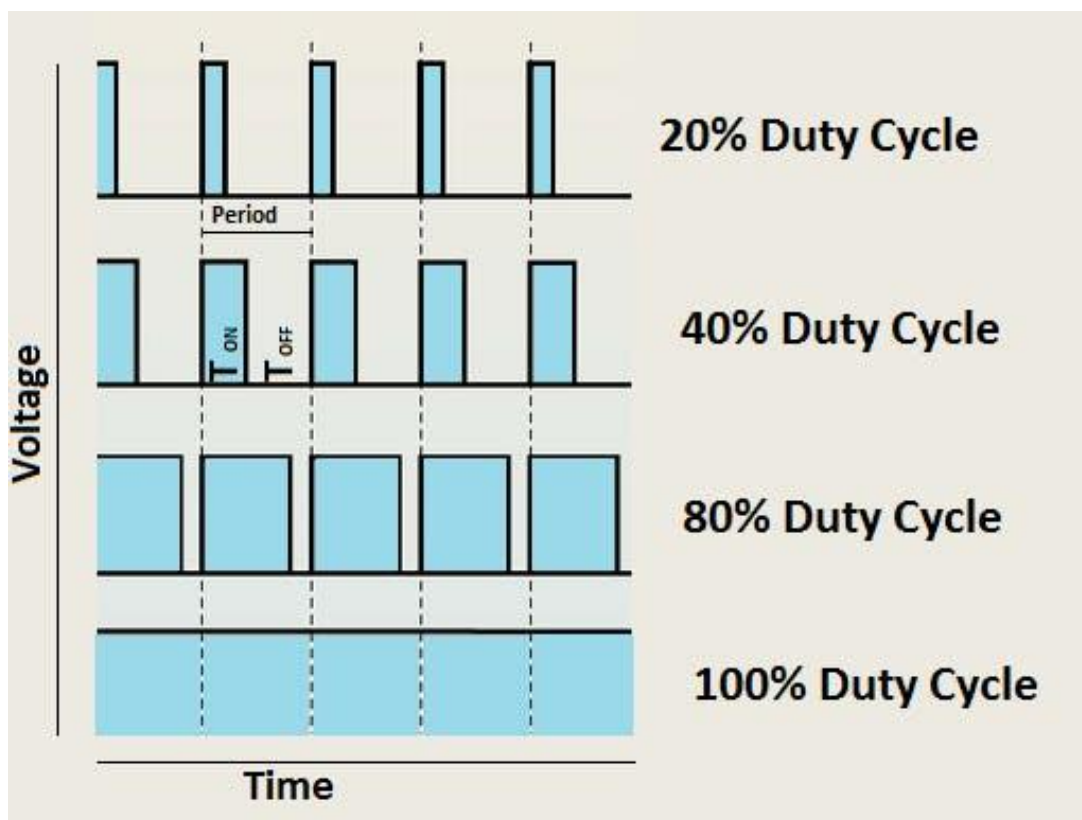
delay(100);
}
```

توضیحات برنامه:

برنامه ابتدا soundLevel را از خروجی Envelope Detector می خواند و با تغییر مقیاس آن را به motorLevel تبدیل می کند. کنترل سرعت موتور DC با استفاده از سیگنال PWM انجام می شود.

### خروجی تنظیم دور موتور با استفاده از PWM

سیگنال PWM در حقیقت یک Square Wave با فرکانس ثابت (۴۹۰۰۰) و Duty Cycle متغیر می باشند (نسبت onTime به offTime) و بدین ترتیب سرعت حرکت موتور DC با تنظیم Duty Cycle تنظیم می شود.



از آنجا که خروجی های **Arduino** می توانند حداکثر ۵ میلی آمپر جریان تامین کنند، برای راه اندازی موتور DC نیاز به یک ترانزیستور می باشد، که بتواند جریان مورد نیاز موتور را تامین کند. به علاوه برای جلوگیری از ورود جریان معکوس هنگام قطع جریان در موتور (ناشی از قانون لنز)، یک دیود در بایاس معکوس با دوسر موتور موازی می شود تا در برابر جریان معکوس از قطعات حساس (ترانزیستور و آردوینو) محافظت کند.

در نهایت مدار کلی به شکل زیر خواهد بود. در اسیلوسکوپ کانال زرد رنگ ورودی خام صدا، ورودی آبی مربوط به خروجی **Envelope Detection Circuit** بوده و دو کانال دیگر به ترتیب سیگنال **PWM** و معکوس آن (کلکتور ترانزیستور درایور موتور) می باشند.

