PWM de alta frequência para gerar senóide de 60 Hz.

Por Eng. Roberto Bairros dos Santos

www.bairrospd.com

Data: 12/07/2017

Sumário

Prefácio.

Este trabalho vai mostrar como montar um programa para Arduino UNO de forma a gerar uma senóide quase sem interferências.

Para gerar a senóide você verá como alterar os temporizadores da placa Arduino UNO de forma a operar o PWM com frequência mais altas.

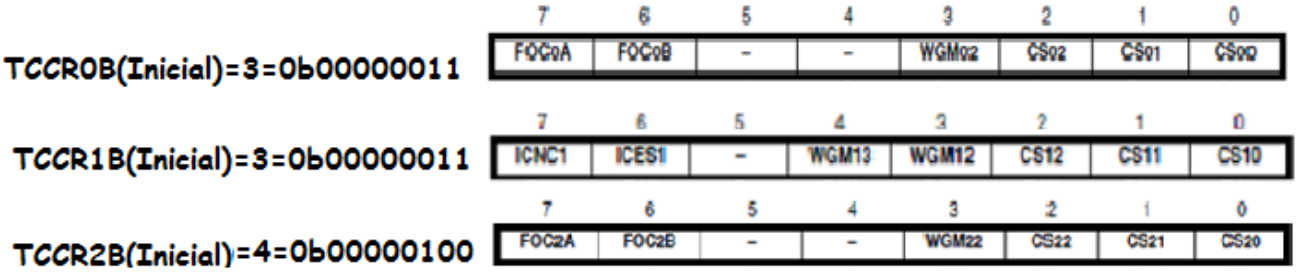
Para você entender melhor os aspectos teóricos sugiro que entre no site www.bairrospd.com e veja a biblioteca com os tutoriais e pdfs que descreve como funcionam os temporizadores do microcontrolador AtMega e como estes temporizadores são usados na placa Arduino UNO.

Como alterar a frequência do PWM.

A forma mais simples consiste em alterar somente os bits de configuração da frequência.

Este é o método mais simples, você só irá mudar os bits CS0x, CS1x e CS2x de qualquer um dos temporizadores TCCRnB!

O valor padrão para cada um dos temporizadores da placa Arduino UNO é mostrado abaixo.



A tabela abaixo mostra um resumo com todas as possibilidades de alterações das frequências dos temporizadores usados para gerar o PWM na placa Arduino UNO.

Esta tabela será útil para alterar a frequência, a linha em amarelo mostra o valor inicial!

TCCR0B=3=0b00000011

TIMER 0 Fast PWM

CS	N	fOCnxPWM	t(ms)
1	1	62500	0,016
2	8	7812,5	0,128
3	64	976,5625	1,024
4	256	244,140625	4,096
5	1024	61,03515625	16,384

$$f_{OCnxPWM} = \frac{f_{clk\_I/O}}{N \cdot 256}$$

Pinos 05 e 06

TCCR1B(Inicial)=3 =0b00000011

TIMER 1 PWM, phase correct

CS	N	fOCnxPCPWM	t(ms)
1	1	31372,55	0,031875
2	8	3921,57	0,255
3	64	490,20	2,040
4	256	122,55	8,160
5	1024	30,64	32,640

$$f_{OCnxPCPWM} = \frac{f_{clk\_I/O}}{2 \cdot N \cdot TOP}$$

Pinos 09 e 10

TCCR2B (Inicial) =4=0b00000100

TIMER 2 PWM, phase correct

CS	N	fOCnxPWM	t(ms)
1	1	31372,54902	0,031875
2	8	3921,568627	0,255
3	32	980,3921569	1,02
4	64	490,1960784	2,04
5	128	245,0980392	4,08
6	256	122,5490196	8,16
7	1024	30,6372549	32,64

$$f_{OCnxPCPWM} = \frac{f_{clk\_I/O}}{N \cdot 510}$$

Pinos 11 e 03

Note que o temporizador 2 é aquele que possui mais opções de frequência, os outros temporizadores trabalham com frequências mais altas.

Com escrever em linguagem “C” a alteração dos bits de frequência.

Existem várias maneiras de escrever em linguagem “C” as alterações dos bits “CSn” do registrador TCCRnB.

Uma forma bem simples é mostrada abaixo onde foi usada a técnica de criar uma máscara com a função “AND” e depois combinar com a função “OR” para chegar ao valor final.

Na máscara com a função “AND” os bits do operador iguais a um são aqueles que não são alterados e os bits iguais zero são zerados.

Na operação final com a função “OR” ocorre o contrário, os bits do operador igual a zero não são alterados, somente os bits iguais a um são alterados!

PWM de alta frequência para gerar senóide.

O exemplo abaixo mostra como alterar o temporizador 1 para a frequência de 3921 Hz.

Estas alterações serão sentidas no pinos 9 e pino 10.

No exemplo deste trabalho vamos usar o pino 9.

O valor marcado como inicial é aquele padrão da placa Arduino UNO, o valor final é aquele com o valor do Temporizador 1 alterado para 3921,57 Hz!

TCCR1B(Inicial)=3 =0b00000011

TCCR1B(final)=2 =0b00000010

TIMER 1 PWM, phase correct

CS	N	fOCnxPCPWM	t(ms)
1	1	31372,55	0,031875
2	8	3921,57	0,255
3	64	490,20	2,040
4	256	122,55	8,160
5	1024	30,64	32,640

$$f_{OCnxPCPWM} = \frac{f_{clk\_I/O}}{2 \cdot N \cdot TOP}$$

Final

Inicial

Pinos 09 e 10

Veja os detalhes da instrução que altera o Temporizador 1.

Primeiro a instrução monta uma máscara que zera os três últimos dígitos, os bits restantes não são alterados.

```
//ajuste do temporizador 1 da saida 9 para 3kHz
TCCR1B=(TCCR1B & 0b11111000) | 0x02;
```

```
                    0b00000101 TCCR2B inicial 490 Hz
                AND 0b11111000
(TCCR1B & 0b11111000) ➪ 0b00000000
```

Na segunda parte uma função OR força os três últimos bits ao valor desejado, neste caso o valor é 2 !

No exemplo o valor foi escrito em hexadecimal.

```
(TCCR1B & 0b11111000) ➪ 0b00000000
                        0b00000010 OR
            | 0x02;   ➪ 0b00000010  TCCR2B final 3921 Hz
```

```
//ajuste do temporizador 1 da saida 9 para 3kHz
TCCR1B=(TCCR1B & 0b11111000) | 0x02;
```

```
                    0b00000101 TCCR2B inicial 490 Hz
                    0b11111000 AND
(TCCR1B & 0b11111000) ➪ 0b00000000
                        0b00000010 OR
            | 0x02;   ➪ 0b00000010  TCCR2B final 3921 Hz
```

Você pode escrever o número 2 em decimal, binário ou hexadecimal.

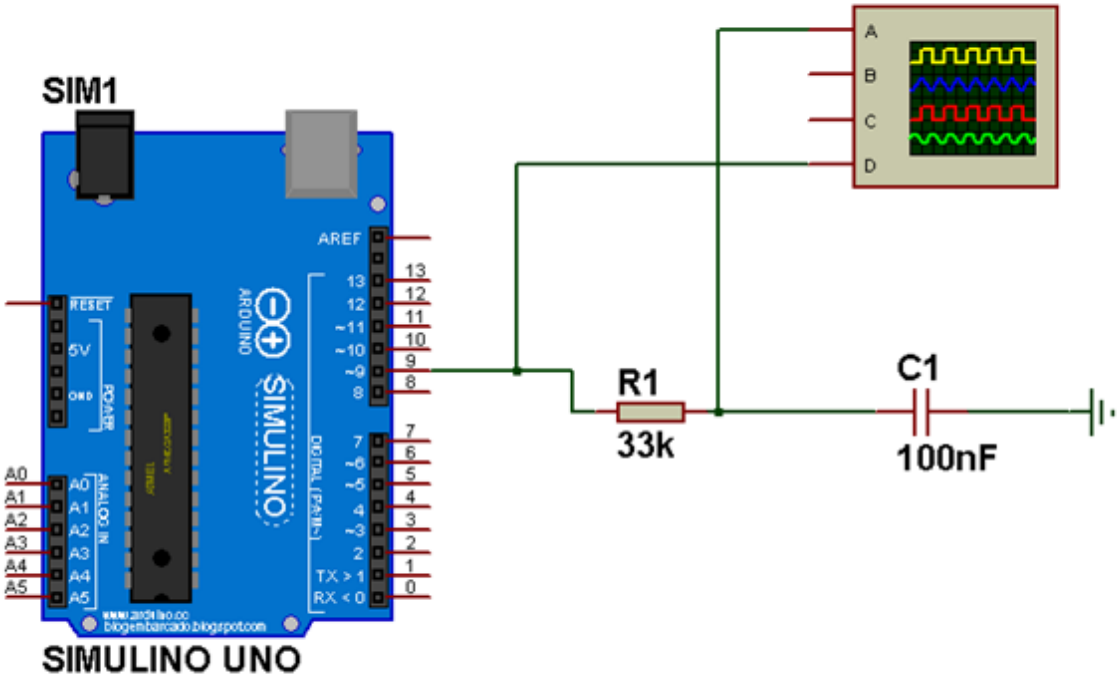
```
TCCR1B=(TCCR1B & 0b11111000) | 2;
TCCR1B=(TCCR1B & 0b11111000) | 0b00000010;
 TCCR1B=(TCCR1B & 0b11111000) | 0x02;
```

O circuito de teste.

O circuito de teste é mostrado abaixo, o osciloscópio serve para você ver a senóide.

A ponteira amarela do osciloscópio está ligada no filtro passa baixo composto por R1 e C1 para transformar o PWM em uma senóide.

A ponteira verde está ligada diretamente a saída 9 do Arduino mostrando o PWM gerado.



Programa simples para gerar a senóide de 60 Hz.

O programa sugerido consiste em criar uma função que gere uma matriz com todos os valores de amplitude e depois na função loop() usar esta matriz para ajustar o PWM a cada ciclo.

A função para gerar a matriz é mostrada abaixo.

A amplitude foi gerada pensando em uma frequência de PWM de 3921Hz no pino 9, para isto o Temporizador 1 deverá ser alterado para CS igual a dois como foi descrito antes.

Uma senóide de 60Hz apresenta um período de 16,7 ms e a frequência de PWM de 3921Hz tem um período de 0,255ms, assim é possível dividir a senóide de 60Hz em 62 passos do PWM!

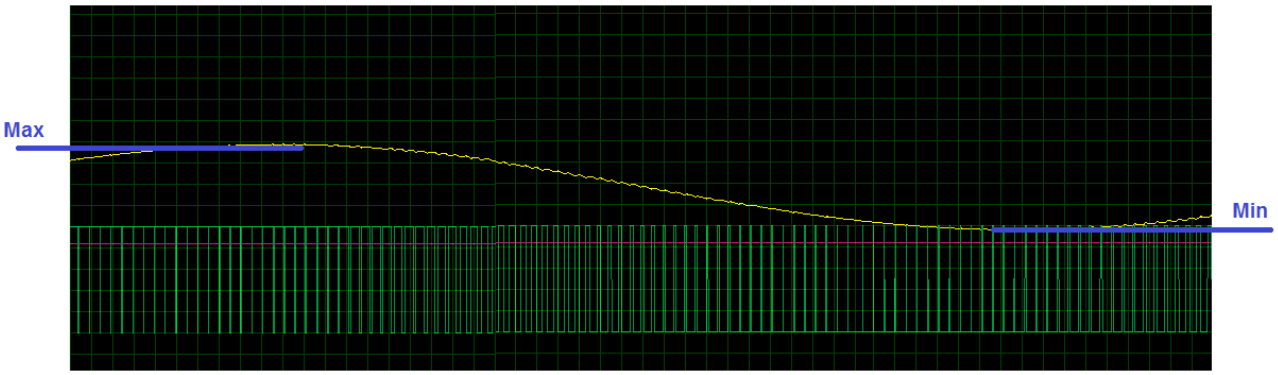
$$\text{Passos} = \frac{16\text{ms}}{0{,}255\text{ms}} = 62$$

Em cada passo o valor do PWM deverá gerar uma amplitude na proporção de uma senóide entre 0 e 5V.

Os valores do PWM devem variar entre 0 e 250, por segurança no programa eu fiz os valores variarem entre 10 e 240 para evitar distorcer a forma de onda.

A figura abaixo mostra em verde o sinal de PWM gerado e em amarelo a senóide, notar que o valor mínimo e máximo de senóide não alcançam os limites 0V e 5V do Arduino.

Esta senóide foi dividida em 62 pulsos de PWM!



PWM de alta frequência para gerar senóide.

O ângulo de passo mínimo é de 5,7375 Graus, assim cada passo do PWM ajusta o valor da amplitude de forma a ajustar o valor da senóide a cada 5,7375 graus.

$$\text{Angulo Mínimo} = \frac{360\ °}{62} = 5{,}7375\ °$$

Observar que na função que gera o seno da linguagem “C” o ângulo é dado em PI Radianos, assim é preciso converter o ângulo mínimo para PI Radianos para ser usado na função.

$$\text{Angulo Minimo em Pi Radianos} = \text{Angulo Minimo em Graus} * PI() / 180$$

$$0{,}1 = 5{,}7375\ ° * 3{,}141516 / 180\ °$$

A rotina consiste em um loop for onde a variável “K” conta os passos e vai ajustando o valor da amplitude.

O valor da amplitude deve ser ajustado em função do seno do ângulo a cada passo, o ângulo a cada passo é o produto do número do passo multiplicado pelo ângulo mínimo em PI Radianos. O valor 10 é o valor mínimo que eu estabeleci para a amplitude e o valor 240 é o valor máximo.

A função é mostrada abaixo.

```
void matrix (){
  //calcula os valores do pwm com amplitude entre 10 e 240
  //deixando uma pargem de 10 por segurança
  for (k=0; k<=62; k++){
    //o valor minimo é 10 e o máximo é 240 convertido para inteiro
    //mais 1 para não ficar negativo
    anguloPi=k*anguloMinimoPi;
    amplitude[k]=int(10+(sin(anguloPi)+1)*240/2);
  }
}
```

O Setup() é mostrado abaixo, onde o pino 9 é definido para o pino da saída e a função geradora da matriz é chamada gerando os valores da amplitude.

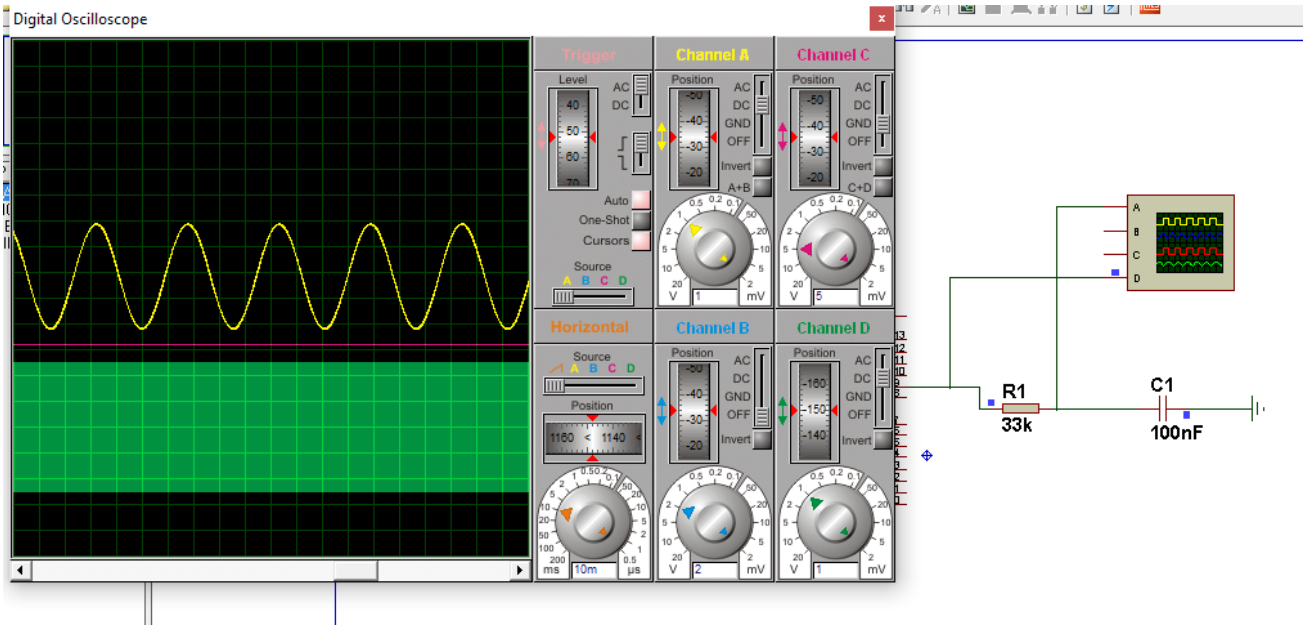
```
29 void setup() {
30   pinMode(pin,OUTPUT);//saída 9 usada como PWM
31  //ajuste do temporizador 2 da saida 9 para 3kHz
32   matrix();//gera a matriz para ser usada mais tarde
33 }
34
```

Finalmente a função de loop() é mostrado abaixo.

Um laço do tipo “for” fica atualizando o valor da saída PWM do pino 9 conforme os valores previamente calculados e guardados na matriz!

```
35 void loop() {
36   // gera a senoide via pwm!
37   for (k=0; k<=62; k++){
38     analogWrite(pin, amplitude[k]); //ajusta pwm
39     delayMicroseconds(T);//tempo do passo
40   }
41
42 }
```

O resultado é mostrado abaixo, mostrando uma senoide quase perfeita!



PWM de alta frequência para gerar senóide.

Na versão pdf você pode copiar e colar o programa descrito abaixo.

```
/**
 *
 *  GERADOR DE SENO COM PWM EM 3KHz
 *
 * 1 ciclo a 60Hz tem periodo de aproximadamente 16,667ms
 *  Frequencia 3921Hz tem periodo de 0,255ms
 *  Numero de passos da senoide. 16,667ms/ 0,255ms = 62 passos
 *  angulo minimo 360/62= 5,806452 ° =0,10134 PIRadianos
 */

int pin=9;//PWM
int T=555;// tempo de 255us para novo ajuste do pwm passo 62
float anguloMinimoPi=0.1;//=angulo minimo em graus=5.73 angulo minimo em pi radianos=0.1
float anguloPi;
int amplitude [120];
int k;//contador do laço

void matrix (){
  //calcula os valores do pwm com amplitude entre 10 e 240
  //deixando uma pargem de 10 por segurança
  for (k=0; k<=62; k++){
    //o valor minimo é 10 e o máximo é 240 convertido para inteiro
    //mais 1 para não ficar negativo
    anguloPi=k*anguloMinimoPi;
    amplitude[k]=int(10+(sin(anguloPi)+1)*240/2);
  }
}

void setup() {
  pinMode(pin,OUTPUT);//saída 9 usada como PWM
 //ajuste do temporizador 2 da saida 9 para 3kHz
  matrix();//gera a matriz para ser usada mais tarde
}

void loop() {
  // gera a senoide via pwm!
  for (k=0; k<=62; k++){
    analogWrite(pin, amplitude[k]); //ajusta pwm
    delayMicroseconds(T);//tempo do passo
  }

}
```

Conclusão.

Você viu como gerar uma senóide usando a saída PWM do Arduino UNO com a frequência do temporizador alterado para 3921Hz gerando uma senóide quase perfeita!

Você pode usar este conceito para construir um inversor, um gerador de sinais e tantas outras aplicações.

Referências.

Bibliografia Auxiliar.

Manual da Atmel para o microcontrolador ATmega328P.

PDF:

Sites: www.bairrospd.com

SEO: www.bairrospd.com, Arduino, PWM, alterando a frequência do PWM, gerando senóide no Arduino, LED, eletrônica, tutorial