# Como alterar a frequência do PWM da placa Arduino UNO.

Por Eng. Roberto Bairros dos Santos

www.bairrospd.com

Data: 07/07/2017

# Sumário

Prefácio	3
O circuito e o programa para testar as alterações.	4
Como alterar a frequência alterando somente os bits de configuração da frequência.	5
Como alterar a frequência alterando a configuração original dos temporizadores e os bits de frequência	10
Conclusão:	23
Referências.	24

# Prefácio.

Este tutorial mostra como alterar a frequência de operação da função analogWrite() da placa Arduino UNO!

Existem duas formas:

- Mantendo a configuração original dos temporizadores e só alterando os bits de ajuste da frequência.
- Alterando a configuração original dos temporizadores e os bits de frequência.

Você deve estar ciente que ao mudar a configuração dos temporizadores algumas funções de tempo do Arduino podem não funcionar corretamente!

Para você entender melhor este tutorial é recomendável ver os tutoriais no site <u>www.bairrospd.com</u> que explicam como funciona o temporizador do microcontrolador ATMega e como medir as frequências dos temporizadores da placa Arduino UNO na sua configuração padrão.

# O circuito e o programa para testar as alterações.

O diagrama abaixo mostra como medir a frequência dos temporizadores.

O osciloscópio foi ligado para mostrar uma saída de cada temporizador.

Neste trabalho foi usado o programa ISIS do PROTEUS com o simulador de Arduino SIMULINO.

Para ver o resultado no terminal da serial foi usado o terminal do Proteus.

O osciloscópio está ligado nos pinos 6 (TIMERO) cor amarelo, 9 (TIMER1) cor azul, 3(TIMER2), cor vermelho!



## Como alterar a frequência alterando 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.

	7	6	5	4	3	2	1	0
TCCR0B(Inicial)=3=0b00000011	FQC0A	FOCoB	-	-	WGM02	CS02	CS01	CSOD
	-							
	1	6	5	4	3	2	1	0
TCCR1B(Inicial)=3=0b0000011	ICNC1	ICES1	-	WGM13	WGM12	C\$12	CS11	CS10
	7	6	5	4	3	2	1	٥
TCCR2B(Inicial)=4=0600000100	F0C2A	F0C2B	-	-	WGM22	CS22	CS21	CS20

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!



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

Existem várias maneiras de escrever em linguagem "C" as alterações dos bits "CSn" do registrador TSSRnB.

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" onde 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!

Para esta alteração você pode escrever o número no formato binário ou decimal.

O exemplo abaixo mostra como alterar o temporizador 2 para a frequência de 3921 Hz.

### TCCR2B (Inicial) =4=0b00000100 TCCR2B(3921Hz)=2=0b00000010

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

Pinos 11 e 03

Veja como escrever a alteração usando número inteiro.

O você pode escrever o número 2 em binário ou hexadecimal.

TCCR2B=(TCCR2B & 0b11111000) | 0x02; TCCR2B=(TCCR2B & 0b11111000) | 0b00000010; )



Programa exemplo.

O circuito para teste é mostrado abaixo.





O programa exemplo abaixo altera a frequência do pino 03 para 3921 Hz Temporizador 2 onde o CS foi alterado usando o valor inteiro, que me parece o mais prático!

```
echo
 1 int numero; int k; int n; char sbuffer[255];
 2 void setup() {
 3
    // put your setup code here, to run once:
    // initialize serial communication at 9600 bits per second:
 4
 5 Serial.begin(9600);
 6 pinMode(3,OUTPUT);
 7
    TCCR2B=(TCCR2B & Ob11111000) | 2;//altera a frequência do temporizador 2 canais 11 e 03 para 3921Hz
 8 }
 9 void loop()
10 // put your main code here, to run repeatedly:
     analogWrite(3,127);
11
12
    if(Serial.available())//se recebeu alguma coisa via serial
13
14
     {
15
      for (k=0;k<255;k++) {sbuffer[k]=0;}//limpa o buffer</pre>
      numero=<mark>Serial.readBytesUntil(13,sbuffer,255);</mark>//le o dado da entrada serial até receber o ENTER
16
      Serial.print("TCCR0A=");Serial.println(TCCR0A); Serial.print("TCCR0B=");Serial.println(TCCR0B);
17
18
      Serial.print("TCCR1A=");Serial.println(TCCR1A);Serial.print("TCCR1B=");Serial.println(TCCR1B);
      Serial.print("TCCR2A=");Serial.println(TCCR2A);Serial.print("TCCR2B=");Serial.println(TCCR2B);
19
20
      Serial.println();Serial.setTimeout(30000);delay(1);
21
     }
22 }
```

int numero;int k;int n;char sbuffer[255]; void setup() { // put your setup code here, to run once: // initialize serial communication at 9600 bits per second: Serial.begin(9600); pinMode(3,OUTPUT); TCCR2B=(TCCR2B & 0b11111000)| 2;//altera a frequência do temporizador 2 canais 11 e 03 para 3921Hz void loop() { // put your main code here, to run repeatedly: analogWrite(3,127); if(Serial.available())//se recebeu alguma coisa via serial { for (k=0;k<255;k++){sbuffer[k]=0;}//limpa o buffer

```
numero=Serial.readBytesUntil(13,sbuffer,255);//le o dado da entrada serial até receber o ENTER
Serial.print("TCCR0A=");Serial.println(TCCR0A); Serial.print("TCCR0B=");Serial.println(TCCR0B);
Serial.print("TCCR1A=");Serial.println(TCCR1A);Serial.print("TCCR1B=");Serial.println(TCCR1B);
Serial.print("TCCR2A=");Serial.println(TCCR2A);Serial.print("TCCR2B=");Serial.println(TCCR2B);
Serial.println();Serial.setTimeout(30000);delay(1);
```

```
}
}
```

}

Para copiar e colar.



O resultado da medição o pino 3 é mostrado na figura, note que eu continuo mostrando todos os registradores, mas a frequência do pino 3 é muito mais lata do que a dos outros pinos!

Como alterar a frequência alterando a configuração original dos temporizadores e os bits de frequência.

Este método é mais amplo você obtém mais possibilidades de valores de frequência, mas tem que alterar completamente a configuração normal dos temporizadores.

A forma normal dos temporizadores é mostrada abaixo, somente o Temporizador O usa o modo "Fast PWM"!

Os modos possíveis de operar são muitos e fogem ao escopo deste trabalho mostrar todos, assim vamos mostrar somente aqueles que fazem os temporizadores funcionar no modo "Fast PWM' e "PWM, phase correct", sem alterar os bits COMnx!

A figura mostra os registradores dos temporizadores com seu valor normal para atuar na instrução analogWrite()!

TCC0A(Inicial) =163=0b10100011	COM0A1	COMOAD	COM0B1	COMoBo	-	- W	GM01 WGM	00
	1	0	1	0	0	0 1	L 1	_
	7	6	5	4	3	2	1	0
TCCROB(Inicial) =3=0b00000011	FOCOA	FOCOB	-	-	WGM02	CS02	CS01	C:500
	í	ō	1	0	0	0	1	1
TCCR2A (Tricial) = 161=0b10100001	COM2A1	COM2A0	COM2B1	T COM2B0	-	4	WGM21	WGM20
					-			
	1	0	1	0	0	_ 0	_ 0	1
	7	6	5	4	3	2	1	0
1 CCK2B (Inicial) =4=000000100	FOC2A	FOC2B	-	-	WGM22	CS22	CS21	CS20
	0	0	0	0	0	1	0	0
TCOD1 4(T-1-1-1) - 1 (1 - 0-101000001	COMIAI	COMIAA	COMIN	COMIN		_	WGM11	WGM10
1CCR1A(Inicial)=161=00101000001	COMTAT	COMINO	COMIBI	COMING	-	-	WOMT	Wamio
	1	0	1	0	0	0	0	1
	7	6	5	4	3	2	1	0
TCCR1B(Inicial)=3 =0b00000011	ICNC1	ICES1	-	WGM13	WGM12	CS12	CS11	CS10
	0	0	0	0	0	0	1	1

11

As tabelas mostram como programar os temporizadores nos dois modos.

TIMER 0

Começando pelo TIMERO, neste temporizador o modo Fast PWM deve ser o preferido.

Mode	WGM02	WGM01	WGM00	Timer/Counter Mode of Operation	тор	Update of OCRx at	TOV Flag Set on <sup>(1)(2</sup>
0	0	0	0	Normal	0xFF	Immediate	MAX
1	0	0	1	PWM, Phase Correct	0xFF	TOP	BOTTOM
2	0	1	0	стс	OCRA	Immediate	MAX
3	0	1	1	Fast PWM	0xFF	BOTTOM	MAX
4	1	0	0	Reserved	-	-	-
5	1	0	1	PWM, Phase Correct	OCRA	TOP	BOTTOM
6	1	1	0	Reserved	-	-	-
7	1	1	1	Fast PWM	OCRA	BOTTOM	TOP

		3	6	5	-4	3	2	đ	0	
TCCOA (PWM, phase correct) = (	60 E	COM0A1 C	OM0A0	COM0B1	COMoBo	-	-	WGM01	WGM00	
		1	0	1	0	0	0	0	0	
	_	7	6	5	- 4	3	2	8	0	
		FOC0A F	OCOB	-	-	WGM02	CS02	CS01	CS00	
TCCR0B (PWM, phase correct)		0	0	0	0	1	×	×	×	
	7	6	5	4	3	2	4	0	_	
TCCOA (FOST PWM) = 63	COM0A	1 COM0A0	COMO	B1 COM0	- 08	-	WGM	01 WGM0	0	
	1	0	1	0	0	0	1	1		
_	7	6	5	4	3	2	: 1	0		
TCCROB (Fast PWM)	FOC0A	FOCoB	-	-	WGN	102 CS	02 CS	01 CS0	0	
	0	0	0	C	0 0	) >	<del>د ا</del>	× 3	×	

### Table 15-9. Clock Select Bit Description

cs	CS02	CS01	CS00	Description					
	0	0	0	No clock source (Timer/Counter stopped)					
1	0	0	1	clk <sub>I/O</sub> /(No prescaling)					
2	0	1	0	clk <sub>I/O</sub> /8 (From prescaler)					
3	0	1	1	clk <sub>I/O</sub> /64 (From prescaler)					
4	1	0	0	clk <sub>I/O</sub> /256 (From prescaler)					
5	1	0	1	clk <sub>I/O</sub> /1024 (From prescaler)					
	1	1	0	External clock source on T0 pin. Clock on falling edge.					
	1	1	1	External clock source on T0 pin. Clock on rising edge.					

### TIMER 0 Fast PWM

Pinos 05 e 06

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

$$f_{OCnxPWM} = \frac{J_{\text{clk}} I/O}{N \cdot 256}$$

TIMER 0	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
		former	1

$$f_{OCnxPCPWM} = \frac{V_{OK}}{N} \cdot 510$$

No Timer 1 o PWM de 8 bits é o mais prático!

Os modos escolhidos são aqueles onde o valor do TOP é o máximo 0xFF (255).

A tabela abaixo mostra como escolher os registradores onde os valores indicados por "x" devem ser preenchidos conforme o de CSnx da pré-escala! 

		WGM12	WGM11	WGM10	Timer/Co	unter Mor	ie of		Und	ate of	TOV1 Ek
Mode	WGM13	(CTC1)	(PWM11)	(PWM10)	Operation	n	201	ТОР	OCR	1X at	Set on
0	0	0	0	0	Normal			0xFFFF	Imme	diate	MAX
1	0	0	0	1	PWM, Ph	ase Corre	st, 8-bit	0x00FF	TOP		BOTTON
2	0	0	1	0	PWM, Ph	ase Corre	rt, 9-bit	0x01FF	TOP		BOTTOM
3	0	0	1	1	PWM, Ph	ase Corre	rt, 10-bit	0x03FF	TOP		BOTTOM
4	0	1	0	0	CTC			OCR1A	Imme	diate	MAX
5	0	1	0	1	Fast PWN	1 <mark>, 8-bit</mark>		0x00FF	BOT	TOM	TOP
6	0	1	1	0	Fast PWN	1, 9-bit		0x01FF	BOT	ГОМ	TOP
7	0	1	1	1	Fast PWN	1, 10-bit		0x03FF	BOT	ГОМ	TOP
8	1	0	0	0	PWM, Ph Correct	ase and Fi	equency	ICR1	BOT	ГОМ	BOTTON
9	1	0	0	1	PWM, Ph Correct	ase and Fi	equency	OCR1A	BOT	ГОМ	BOTTON
10	1	0	1	0	PWM, Ph	ase Corre	t	ICR1	TOP		BOTTON
11	1	0	1	1	PWM, Ph	ase Corre	rt	OCR1A	TOP		BOTTON
12	1	1	0	0	CTC			ICR1	Imme	ediate	MAX
13	1	1	0	1	(Reserved	j)		-	-		-
14	1	1	1	0	Fast PWN	1		ICR1	BOT	ГОМ	TOP
15	1	1	1	1	Fast PWN	4		OCR1A	BOT	ТОМ	TOP
				т		5	a	я	-7.	31	
TCCP1	A (PWM	phase co	rrect) = 14	1 COM1A1	CONTAC	COM1B1	COM1B0	-	-	WGM1	1 WGN1
			,,	1	0	1	0	0	0	0	1
				7	6	5	4	8	2	1	0
TCCR	1B (PWM	, phase c	orrect)	ICNC1	ICES1	-	WGM13	WGM12	C612	CS11	CS10
				0	0	0	0	0	×	×	×
TCCP	14 (East	PW/M) -	161	7	6	5	4	3	2	1	0
rook				COMIAI	CONTAC	COMIN	COMIBO	-	-	WGM	i wan
				1	0	1	0	0	0	0	1
TOO	10 (East	P\A/AA)		7	6	5	4	3	2	1	0

### Table 16-5. Clock Select Bit Description

cs

### -----0010 ----

s	CS12	CS11	CS10	Description					
	0	0	0	No clock source (Timer/Counter stopped).					
1	0	0	1	clk <sub>VD</sub> /1 (No prescaling)					
2	0	1	0	clk <sub>VO</sub> /8 (From prescaler)					
3	0	1	1	clk <sub>VO</sub> /64 (From prescaler)					
4	1	0	0	clk <sub>VO</sub> /256 (From prescaler)					
5	1	0	1	clk <sub>VO</sub> /1024 (From prescaler)					
	1	1	0	External clock source on T1 pin. Clock on falling edge.					
	1	1	1	External clock source on T1 pin. Clock on rising edge.					

### TIMFR 1 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

 $f_{OCnxPWM} = \frac{f_{clk\_I/O}}{N \cdot (1 + TOP)}$ 

### 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				
$f_{OCnxPCPWM} = \frac{f_{clk}   v \hat{O}}{2 \cdot N \cdot TOP}$							

Por fim o TIMER2, neste temporizador o FAst PWM deve ser o modo preferido.

### Pinos 11 e 03 TIMER 2

Table 18-8. Waveform Generation Mode Bit Description

Mode	WGM2	WGM1	WGM0	Timer/Counter Mode of Operation	тор	Update of OCRx at	TOV Flag Set on <sup>(1)(2)</sup>	
0	0	0	0	Normal	0xFF	Immediate	MAX	
1	0	0	1	PWM, Phase Correct	0xFF	TOP	BOTITOM	
2	0	1	0	CTC	OCRA	Immediate	MAX	
3	0	1	1	Fast PWM	0xFF	BOTTOM	MAX	
4	1	0	0	Reserved	-	-	-	
5	1	0	1	PWM, Phase Correct	OCRA	TOP	BOTTOM	
6	1	1	0	Reserved	-	-	-	
7	1	1	1	Fast PWM	OCRA	BOTTOM	TOP	
TCCR2	2A (PWM 2B: (PWN	, phase c 1, phase :	orrect) = correct)	7 6 COM2A1 COM2A 1 0 7 6 FOCDA FOCO	5 0 COM281 1 5. 8 -	4 3 COM2B0 - 0 0 4 WC	2 ( - WG 0 ( 3 2 Moz C5oz (	0 M21 WGM20 1 1 1 0 5501 C500
				0 0	0	0 0	×	× ×
TCCR	A (Fast	PWM) =	163	7 6 COM2A1 COM2A0	5 COM2B1	4 3 COM2B0 -	2 1   -   WGM	0 21 WGM20
				1 0	1	0 0	0 1	1
TCCD	B (East			7 6 F000A E000B	5	4 3	2 02 0502 05	0 01 CS00
TUCK	LD: (FUST)	ww.		0 0	0	0 0	×	x x

### Table 18-9. Clock Select Bit Description

C

s	CS22	CS21	CS20	Description
	0	0	0	No clock source (Timer/Counter stopped).
1	0	0	1	clk <sub>T2S</sub> /(No prescaling)
2	0	1	0	clk <sub>T29</sub> /8 (From prescaler)
3	0	1	1	clk <sub>T26</sub> /32 (From prescaler)
4	1	0	0	clk <sub>T2S</sub> /64 (From prescaler)
5	1	0	1	clk <sub>T2S</sub> /128 (From prescaler)
6	1	1	0	clk <sub>T2S</sub> /256 (From prescaler)
7	1	1	1	clk <sub>T25</sub> /1024 (From prescaler)

TIMER 2 Fast PWM

TIVIER	Z rasi	F VV IVI		
CS	N	<b>f</b> OCnxPWM	t(ms)	
1	1	62500	0,016	
2	8	7812,5	0,128	
3	32	1953,125	0,512	
4	64	976,5625	1,024	
5	128	488,28125	2,048	
6	256	244,140625	4,096	
7	1024	61,03515625	16,384	

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

 $f_{OCnxPWM} = \frac{f_{Clk\_VO}}{N \cdot 256}$ 

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

Exemplo 1:

Alterando a frequência do pino 9 para 7812Hz(0,128 ms =128 us) usando o modo "Fast PWM", para isto você deve alterar o Temporizador 1 como descrito abaixo.

TCCRLA (PWM, phase correct) #161	CORRE	COM-M	COMPANY	0.000			1000	<b>BORNE</b>
	1	0	4	0	0	0	0	4
	- W			. 6			1.1	0
TCCR18 (PWM, phase correct)	ISNE?	1000		1000	1000	0.00	004	0000
		0	0		6	-		-
	8	8	- 0	8 - C		8	1	0
TECRIA (PAR PWA) 1161	COMPLEX.	104.4	0.0418	COMPANY.			BOD N	actes.
	1	0	1	0	0	0	0	1
TCCR18 (Fast FWM)	THE P	1000	_		10.000	- 100	104	
	1.000	- Contraction	_		_	10100	10100	
	0	0	0			100	<b>X</b>	18

TIMFR 1 Fast PWM

.016
1-20
,128
,024
,096
,384

CS	N	<b>focnxPCPWM</b>	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
focnx	PCPWM <sup>=</sup>	$= \frac{f_{clk_l} v_0}{2 \cdot N \cdot TOP}$	

TIMER 1 PWM, phase correct

Aqui parece melhor programar em binário.

TCCR1A=0b10100001;//Timer 1 operando no modo "Fast PWM"

TCCR1B=0b00001010;//Frequência de 7812 Hz 0,128ms =128us

TCCR1B=0b00001010; 010 binário=2 decimal

### O programa completo:

```
echo §
 1 int numero; int k; int n; char sbuffer[255];
 2 void setup() {
 3 // put your setup code here, to run once:
    // initialize serial communication at 9600 bits per second:
 4
 5
    Serial.begin(9600);
 6
    pinMode(9,OUTPUT);
     TCCR1A=0b10100001;//Timer 1 operando no modo "Fast PWM
 7
    TCCR1B=0b00001010;//Frequência de 7812Hz 0,128 ms =128 us
 8
 9 }
10 void loop() {
11 // put your main code here, to run repeatedly:
analogWrite(9,127);
13 if(Serial.available())//se recebeu alguma coisa via serial
14 {
         for (k=0;k<255;k++) {sbuffer[k]=0;}//limpa o buffer</pre>
      numero=Serial.readBytesUntil (13, sbuffer, 255);//le o dado da entrada serial até receber o ENTER
16
17
       Serial.print("TCCR0A=");Serial.println(TCCR0A);Serial.print("TCCR0B=");Serial.println(TCCR0B);
18
       serial.print("TCCR1A="); Serial.println(TCCR1A); Serial.print("TCCR1B="); Serial.println(TCCR1B);
       Serial.print("TCCR2A=");Serial.println(TCCR2A);Serial.print("TCCR2B=");Serial.println(TCCR2B);
19
20
       Serial.println();Serial.setTimeout(30000);delay(1);
21 }
22 }
```



Para copiar e colar.

```
int numero;int k;int n;char sbuffer[255];
void setup() {
 // put your setup code here, to run once:
 // initialize serial communication at 9600 bits per second:
 Serial.begin(9600);
 pinMode(9,OUTPUT);
 TCCR1A=0b10100001;//Timer 1 operando no modo "Fast PWM"
 TCCR1B=0b00001010;//Frequência de 7812 Hz 0,128ms =128us
}
void loop() {
 // put your main code here, to run repeatedly:
 analogWrite(9,127);
 if(Serial.available())//se recebeu alguma coisa via serial
 {
  for (k=0;k<255;k++){sbuffer[k]=0;}//limpa o buffer</pre>
  numero=Serial.readBytesUntil(13,sbuffer,255);//le o dado da entrada serial até receber o ENTER
  Serial.print("TCCR0A=");Serial.println(TCCR0A); Serial.print("TCCR0B=");Serial.println(TCCR0B);
  Serial.print("TCCR1A=");Serial.println(TCCR1A);Serial.print("TCCR1B=");Serial.println(TCCR1B);
  Serial.print("TCCR2A=");Serial.println(TCCR2A);Serial.print("TCCR2B=");Serial.println(TCCR2B);
  Serial.println();Serial.setTimeout(30000);delay(1);
 }
}
```

### O Resultado:



8

Exemplo 2:

Alterando a frequência do pino 3 para 7815Hz(0,128 ms =128 us), para isto você deve alterar o Temporizador 2 como descrito abaixo.

	7	6	5	4	3	2	9	0
TCCR2A (PWM, phase correct) =61	COM2A1	COM2A0	COM2B1	COM2B0	-	-	WGM21	WGM20
	1	0	1	0	0	0	0	1
	7	6	6.	4	3	2	1	0
TCCR2B: (PWM, phase correct)	FOCOA	FOCOB	-	-	WGM02	C\$62	C\$01	CS00
	0	0	0	0	0	×	×	×
	7	6	5	4	3	2	1	0
TCCR2A (Fast PWM) = 163	7 COM2A1	6 COM2A0	5 COM2B1	4 COM2B0	3	2	1 WGM21	0 WGM20
TCCR2A (Fast PWM) = 163	7 COM2A1	6 COM2A0 <b>0</b>	5 COM2B1	4 COM2B0 O	8 -   0	2 - 0	1 WGM21   1	0 WGM20 1
TCCR2A (Fast PWM) = 163	7 COM2A1 1 7	6 COM2A0 <b>0</b> 8	5 COM2B1 1 5-	4 COM2B0 0 4	8 - 0 3	2 - 0 2	1 WGM21   1 1	0: WGM20 1
TCCR2A (Fast PWM) = 163 TCCR2B: (Fast PWM)	7 COM2A1 1 7 F0C6A	6 COM2A0 0 6 FOCOB	5 COM2B1 1 6 -	4 COM2B0 0 4 -	3 - 0 3 WGM02	2 - 0 2 C\$02	1 WGM21   1 (5501	0: WGM20 1 0 C500

TIMER	2 Fast	PWM	
CS	N	<b>focnxPWM</b>	t(ms)
1	1	62500	0,016
2	8	7812,5	0,128
3	32	1953,125	0,512
4	64	976,5625	1,024
5	128	488,28125	2,048
6	256	244,140625	4,096
7	1024	61,03515625	16,384

5	2,048	5
5	4,096	6
5	16,384	. 7

$$f_{OCnxPWM} = \frac{J_{clk\_VO}}{N \cdot 256}$$

INVIEN	2	phase correct	1
CS	N	<b>focnxPWM</b>	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

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

TIMER 2 DWM phase correct

TCCR2A=0b10100011;//Timer 2 operando no modo "Fast PWM"

TCCR2B=0b00000010;//Frequência de 7815 Hz 0,128ms =128us

TCCR2B=0b00000010 //Frequência de 7815 Hz 0,128ms =128us

010 binário=2 decimal

### O programa:

```
echo §
 1 int numero; int k; int n; char sbuffer[255];
 2 void setup() {
 3 // put your setup code here, to run once:
 4 // initialize serial communication at 9600 bits per second:
 5
    Serial.begin(9600);
 6
    pinMode(3,OUTPUT);
 7
    TCCR2A=0b10100011;//Timer 2 operando no modo "Fast PWM"
 8
    TCCR2B=0b00000010;//Frequência de 7815 Hz 0,128ms =128us
10 }
11 void loop() {
12 // put your main code here, to run repeatedly:
13 analogWrite(3,127);
14 if(Serial.available())//se recebeu alguma coisa via serial
15 {
      for (k=0;k<255;k++){sbuffer[k]=0;}//limpa o buffer</pre>
16
17
      numero=Serial.readBytesUntil (13, sbuffer, 255); //le o dado da entrada serial até receber o ENTER
       Serial.print("TCCR0A=");Serial.println(TCCR0A); Serial.print("TCCR0B=");Serial.println(TCCR0B);
18
      Serial.print("TCCR1A=");Serial.println(TCCR1A);Serial.print("TCCR1B=");Serial.println(TCCR1B);
19
      serial.print("TCCR2A=");Serial.println(TCCR2A);Serial.print("TCCR2B=");Serial.println(TCCR2B);
20
21
      Serial.println();Serial.setTimeout(30000);delay(1);
22
    }
```

```
Para colar e copiar.
int numero;int k;int n;char sbuffer[255];
void setup() {
 // put your setup code here, to run once:
 // initialize serial communication at 9600 bits per second:
 Serial.begin(9600);
 pinMode(3,OUTPUT);
 TCCR2A=0b10100011;//Timer 2 operando no modo "Fast PWM"
 TCCR2B=0b00000010;//Frequência de 7815 Hz 0,128ms =128us
void loop() {
 // put your main code here, to run repeatedly:
 analogWrite(3,127);
 if(Serial.available())//se recebeu alguma coisa via serial
 {
  for (k=0;k<255;k++){sbuffer[k]=0;}//limpa o buffer</pre>
  numero=Serial.readBytesUntil(13,sbuffer,255);//le o dado da entrada serial até receber o ENTER
  Serial.print("TCCR0A=");Serial.println(TCCR0A); Serial.print("TCCR0B=");Serial.println(TCCR0B);
  Serial.print("TCCR1A=");Serial.println(TCCR1A);Serial.print("TCCR1B=");Serial.println(TCCR1B);
  Serial.print("TCCR2A=");Serial.println(TCCR2A);Serial.print("TCCR2B=");Serial.println(TCCR2B);
  Serial.println();Serial.setTimeout(30000);delay(1);
 }
```

}

}

### O resultado:



.2

# Conclusão:

Você viu como alterar a frequência do PWM do Arduino usando dois métodos, no primeiro você simplesmente altera a frequência; no segundo você altera o modo de gerar o PWM e a frequência.

Referências.

Bibliografia.

Manual da Atmel para o microcontrolador ATmega328P.

PDF:

Sites: www.bairrospd.com

SEO: www.bairrospd.com, Arduino, PWM, alterando a frequência do PWM, senóide, LED, eletrônica, tutorial