As you can see in last two post about sine waves ([How to generate a sine wave from arduino or atmega 328](http://www.eprojectszone.com/2016/08/21/how-to-generate-a-sine-wave-from-arduino-or-atmega-328/) and [How to generate two phase sine waves with arduino](http://www.eprojectszone.com/2017/02/22/how-to-generate-two-phase-sine-waves-with-arduino/)) is not as hard as you think to generate such signals so, in this post i will generate three phase sine wave signals at **120 degrees phase shift**.

Because arduino uno has three timers and one of them is used for interrupt it can not be used anymore so i have used an arduino mega.

Arduino mega has an **atmega 2560** onboard with **five timers and 15 pwm capable pins**.

Let’s see the corresponding between timers and pins:

-Timer 0-pin 4 (OC0B) and pin 13(OC0A)

-Timer 1-pin 11(OC1A) and pin 12(OC1B)

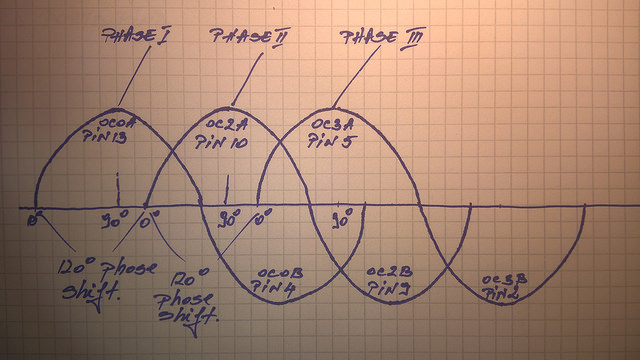
-Timer 2-pin 9(OC2B) and pin 10(OC2A)

-Timer 3-pin 2(OC3B), pin 3(OC3C) and pin 5(OC3A)

-Timer 4-pin 6(OC4A), pin 7(OC4B) and pin 8(OC4C)

-Timer 5-pin 44(OC5C), pin 45(OC5B) and pin 46(OC5A)

For this project i use **Timer 1 for interrupt** like in the previous post **Timer 0 for first phase**, **Timer 2 for the second phase**and **Timer 3 for the third phase** with **OCxA for positive half duty cycle** and **OCxB for the negative half duty cycle** like in the picture below:

[](https://www.flickr.com/photos/144378753@N08/33296404501/in/dateposted-public/)

As you can see in the previous post with two phase, if for **180 degrees** we have **314 elements** , for **120 degrees** we have **209 elements** so the second signal must start when the first is at 209 pulse and the third must start when the second is at 209 pulse.

So, when the program starts, the interrupt is enabled and first is executed the first signal part(with element i). When i take the 209 value the second signal is enabled(with element j). When j take the 209 value the third signal is enabled. In this way these three signal are at 120 degrees phase shift.

To ensure the phase shift we use an “if” function and  for the second wave is like below:

“if ((i==209) || OK1==1){  
OK1=1;”

When “i” has 209th value the “if” function is enabled  and everything in it is executed. To mantain the execution of that part for the second signal after the value of i(209) is changed we use a variable which enables the “if” function continuously.

For the third signal the if function is like below:

“if ((j==209) || OK3==1){  
OK3=1;”

In this case, the third signal is enabled when  “j” (the element for the second signal)  has the 209th value. After that, the “if” function is executed like for the second signal.

Because i have an oscilloscope with only **two channels,**next i will present these signals in pairs.

Some signals will be **inverted** from the oscilloscope software to simulate like they are driving a H bridge on the negative half duty cycle, but reported to arduino GND they still have **5 V**.

As you can see in the picture above i have used **Timer 0 for first signal, Timer 2 for the second and Timer 3 for the third**.

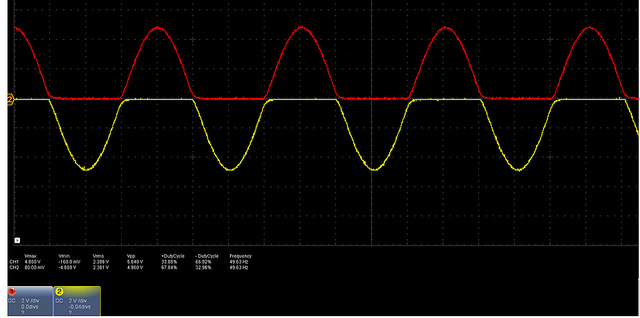
–**Timer 0-pin 4 (OC0B) and pin 13(OC0A)**

–**Timer 2-pin 9(OC2B) and pin 10(OC2A)**

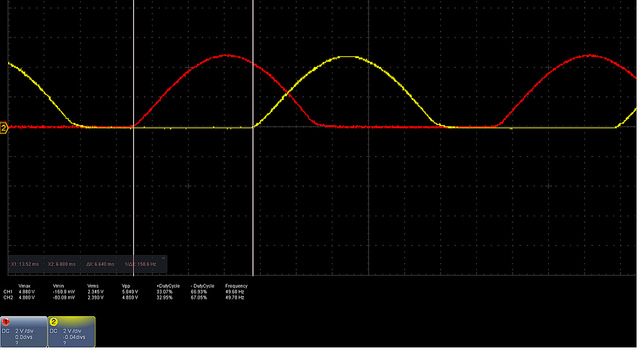
–**Timer 3-pin 2(OC3B), pin 3(OC3C-not used in this example) and pin 5(OC3A)**

First let’s see the first signal.

For that we use OC0A and OC0B **pins(13 and 4)**. On the oscilloscope, the negative half duty cycle(from pin 4) is inverted in order to appear like a negative half duty cycle.

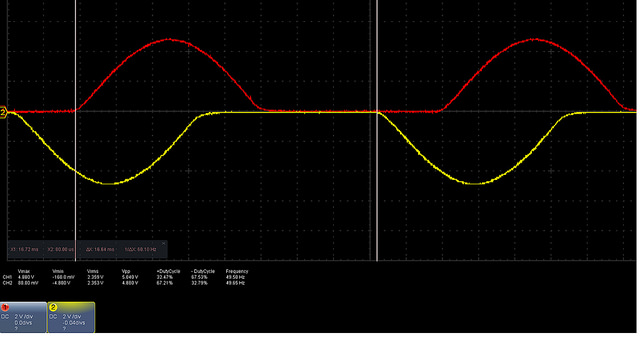
[](https://www.flickr.com/photos/144378753@N08/33408504561/in/dateposted-public/)

In the image below we have the positive half duty cycle of the first and the second signals OCR0A and OCR2A(**pin 13 and pin 10**)

[](https://www.flickr.com/photos/144378753@N08/32693676764/in/dateposted-public/)

The red signal is from pin 13 OCR0A and the yelow one from pin 10. The time delay between them is **6 .640 ms**, almost like 6.66 for 120 degrees phase shift.

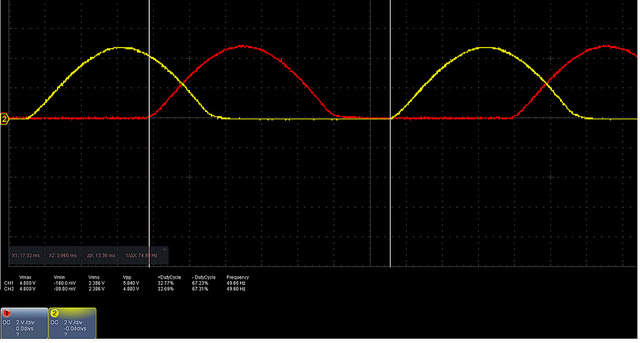
Next is shown the positive half duty cycle of the first signal(**pin 13**) and the negative of the second signal(**pin 9**) inverted:

[](https://www.flickr.com/photos/144378753@N08/33380477472/in/dateposted-public/)

The time delay between the start of the positive duty cycle of the first signal  and the negative of the second signal is **16.64ms.**

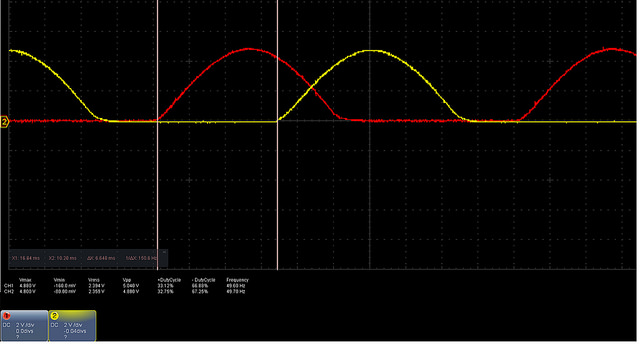
These 16.64ms results from the 6.64ms delay between positive half duty cycles of the first signal and the positive of the second and 10 ms for one half duty cycle(50Hz signal).

The positive duty cycle of the first signal(**pin 13**) and positive of the third(**pin 5**) is:

[](https://www.flickr.com/photos/144378753@N08/33409019881/in/dateposted-public/)

Time delay between these two is **13.36 ms** almost 13.32 for 240 degrees phase shift.

The same duty cycles but from the second(**pin 10**) and third(**pin 5**) signal are:

[](https://www.flickr.com/photos/144378753@N08/32694159634/in/dateposted-public/)

The results are identical with the results between signal 1 and signal 2 also with 6.640ms delay.

The program for this project is:  
int i=0;  
int j=0;  
int k=0;  
int x=0;  
int y=0;  
int z=0;  
int OK=0;  
int OK1=0;  
int OK2=0;  
int OK3=0;  
int OK4=0;  
int sinPWM[]={1,2,5,7,10,12,15,17,19,22,24,27,30,32,34,37,39,42,  
44,47,49,52,54,57,59,61,64,66,69,71,73,76,78,80,83,85,88,90,92,94,97,99,  
101,103,106,108,110,113,115,117,119,121,124,126,128,130,132,134,136,138,140,142,144,146,  
148,150,152,154,156,158,160,162,164,166,168,169,171,173,175,177,178,180,182,184,185,187,188,190,192,193,  
195,196,198,199,201,202,204,205,207,208,209,211,212,213,215,216,217,219,220,221,222,223,224,225,226,227,  
228,229,230,231,232,233,234,235,236,237,237,238,239,240,240,241,242,242,243,243,244,244,245,245,246,246,  
247,247,247,248,248,248,248,249,249,249,249,249,255,255,255,255,249,249,249,249,249,248,  
248,248,248,247,247,247,246,246,245,245,244,244,243,243,242,242,241,240,240,239,238,237,237,236,235,234,  
233,232,231,230,229,228,227,226,225,224,223,222,221,220,219,217,216,215,213,212,211,209,208,207,205,204,  
202,201,199,198,196,195,193,192,190,188,187,185,184,182,180,178,177,175,173,171,169,168,166,164,162,160,  
158,156,154,152,150,148,146,144,142,140,138,136,134,132,130,128,126,124,121,119,117,115,113,110,108,106,  
103,101,99,97,94,92,90,88,85,83,80,78,76,73,71,69,66,64,61,59,57,54,52,49,47,44,42,39,37,34,32,30,  
27,24,22,19,17,15,12,10,7,5,2,1};

void setup() {  
Serial.begin(9600);

pinMode(4, OUTPUT);  
pinMode(13,OUTPUT);  
pinMode(9,OUTPUT);  
pinMode(10,OUTPUT);  
pinMode(2,OUTPUT);  
pinMode(3,OUTPUT);  
pinMode(5,OUTPUT);

cli();// stop interrupts  
TCCR0A=0;//reset the value  
TCCR0B=0;//reset the value  
TCNT0=0;//reset the value  
//0b allow me to write bits in binary  
TCCR0A=0b10100001;//phase correct pwm mode  
TCCR0B=0b00000001; //no prescaler

TCCR2A=0;//reset the value  
TCCR2B=0;//reset the value  
TCNT2=0;//reset the value  
//0b allow me to write bits in binary  
TCCR2A=0b10100001;//phase correct pwm mode  
TCCR2B=0b00000001; //no prescaler

TCCR3A=0;//reset the value  
TCCR3B=0;//reset the value  
TCNT3=0;//reset the value  
//0b allow me to write bits in binary  
TCCR3A=0b10100001;//phase correct pwm mode  
TCCR3B=0b00000001; //no prescaler  
TCCR1A=0;//reset the value  
TCCR1B=0;//reset the value  
TCNT1=0;//reset the value  
OCR1A=509;// compare match value  
TCCR1B=0b00001001; //WGM12 bit is 1 for CTC mode and no prescaler

TIMSK1 |=(1 << OCIE1A);// enable interrupts

sei();//stop interrupts  
}  
ISR(TIMER1\_COMPA\_vect){// interrupt when timer 1 match with OCR1A value  
if(i>313 && OK==0){// final value from vector for pin 13  
i=0;// go to first value of vector  
OK=1;//enable pin 4  
}  
if(i>313 && OK==1){// final value from vector for pin 4  
i=0;//go to firs value of vector  
OK=0;//enable pin 13  
}  
x=sinPWM[i];// x take the value from vector corresponding to position i(i is zero indexed)  
i=i+1;// go to the next position  
if(OK==0){  
OCR0B=0;//make pin 4 0  
OCR0A=x;//enable pin 13 to corresponding duty cycle  
}  
if(OK==1){  
OCR0A=0;//make pin 13 0  
OCR0B=x;//enable pin 4 to corresponding duty cycle  
}

if ((i==209) || OK1==1){//if i equal with 209 enable second signal  
OK1=1;//mantain if function  
if(j>313 && OK2==0){// final value from vector for pin 10  
j=0;// go to first value of vector  
OK2=1;//enable pin 9  
}  
if(j>313 && OK2==1){// final value from vector for pin 9  
j=0;//go to firs value of vector  
OK2=0;//enable pin 10  
}  
y=sinPWM[j];// y take the value from vector corresponding to position j(j is zero indexed)  
j=j+1;// go to the next position  
if(OK2==0){  
OCR2B=0;//make pin 9 0  
OCR2A=y;//enable pin 10 to corresponding duty cycle  
}  
if(OK2==1){  
OCR2A=0;//make pin 10 0  
OCR2B=y;//enable pin 9 to corresponding duty cycle  
}  
}  
if ((j==209) || OK3==1){//if j equal with 209 enable third signal  
OK3=1;//mantain if function  
if(k>313 && OK4==0){// final value from vector for pin 5  
k=0;// go to first value of vector  
OK4=1;//enable pin 2  
}  
if(k>313 && OK4==1){// final value from vector for pin 2  
k=0;//go to firs value of vector  
OK4=0;//enable pin 5  
}  
z=sinPWM[k];// z take the value from vector corresponding to position k(k is zero indexed)  
k=k+1;// go to the next position  
if(OK4==0){  
OCR3B=0;//make pin 2 0  
OCR3A=z;//enable pin 5 to corresponding duty cycle  
}  
if(OK4==1){  
OCR3A=0;//make pin 5 0  
OCR3B=z;//enable pin 2 to corresponding duty cycle  
}  
}  
}  
void loop() {

}