

3 Phase Sine Wave Generator code | Arduino

In this post we are going to construct a circuit using Arduino which can generate 3 phase sine wave with 120 degree phase difference whose frequency can be varied by using a potentiometer. We will be inspecting the generated waveform using an oscilloscope and also with serial plotter of Arduino IDE to see whether we are really generating 3 phase sine wave. There are some key concepts to be explored before we build the circuit and code, so that we can understand the project from core.

Three Phase Inverter Circuit Diagram: [Click here](#)

We will see:

- What is PWM?
- What is SPWM?
- Why 3 phases are with 120 degree phase difference?
- Circuit Diagram
- Program code
- Circuit Description
- Generated wave form images

What is PWM?

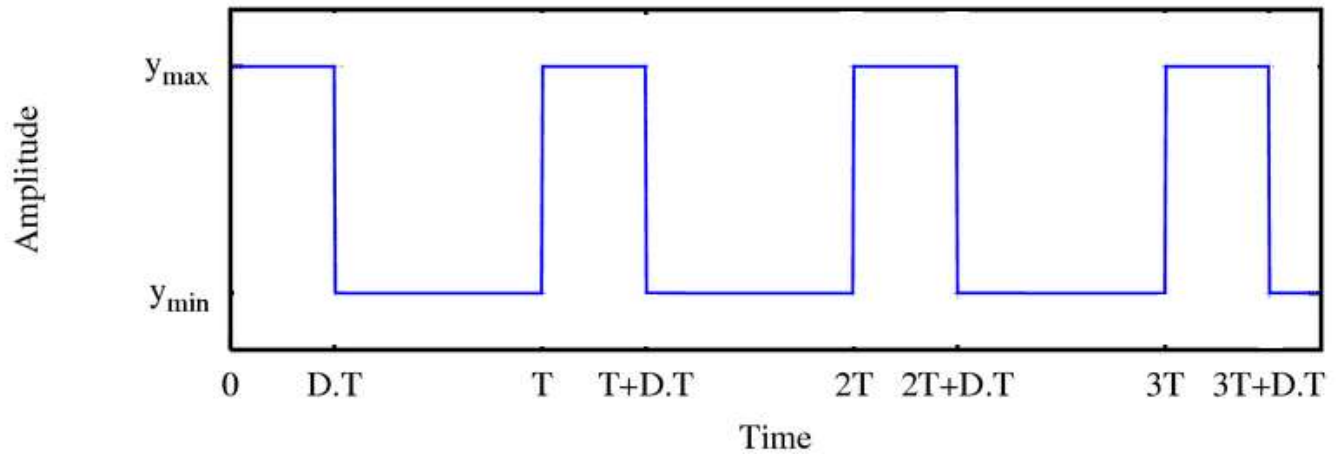
PWM stands for Pulse Width Modulation; it is a type of digital modulation for controlling simple and complex electronic modules and devices. PWM technique is used for making analog voltage levels using digital microcontrollers or microprocessors or ICs.

The Arduino and other microcontrollers are digital devices; they can only understand and also generate two voltage levels: HIGH and LOW. With just two voltage levels we are generating wide range of voltage levels and this technique is called PWM.

How can we generate variable voltage level with just HIGH and LOW digital levels?

To understand PWM more deeply we have to understand a term **“Duty Cycle”**. Duty cycle is represented in percent from 0 to 100%. 0% duty cycle means no voltage at output and 100% duty cycle means maximum voltage of the power source.

Here with microcontroller full voltage is generally 5V, so at 100% duty cycle we get 5V, if we set duty cycle to 50% we will get **average voltage** as half of the source voltage, here it is 2.5V at output.



PWM

As you can see from above image there are 2-levels "Ymax and "Ymin" switched with a specific time period. If the duty cycle of the pulse is 50% we will get 50% ON and 50% OFF and we will get average voltage of 2.5V for 5V voltage source.

If the duty cycle of the pulse is 75%, the Ymax will be at HIGH for 75% of the time and 25% of the time will be LOW at Ymin and we will get average voltage of 3.75V for 5V voltage source.

The Pulse width of the ON period of 75% will be thicker than pulse width of 50%. A square wave with 0% duty cycle will have straight line horizontal at "Ymin", if the duty cycle is 100% we will get a straight line horizontally at "Ymax".

To conclude, we will get higher **average voltage** if the width of the pulse is wider and vice versa. Just by varying the pulse width we can get the desire **average voltage levels** with just two voltage states.

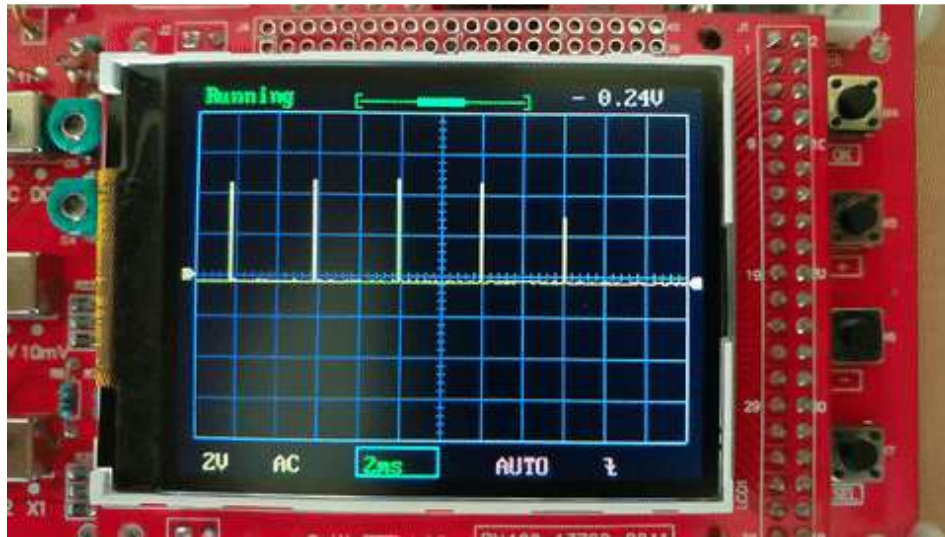
If an incandescent bulb is connected across the output and if we apply pulse at 50% duty cycle the bulb will glow at 50% of its brightness and if we apply pulse width of duty cycle with 25% the bulb will glow at 25% of its brightness. Similarly we can control the speed of a DC motor / LED etc.

What is SPWM?

SPWM stands for Sinusoidal Pulse Width Modulation; it is a kind of PWM technique which generates sinusoidal equivalent waveform.

We know that by varying the thickness of the pulse we can get the desire voltage levels. In SPWM technique the pulses are varied in a predetermined way by using microcontrollers or any electronics circuits in such a way that the waveform across the output is sinusoidal equivalent.

Let's take a look at the SPWM generated by the proposed circuit:



SPWM using Arduino

As we can see that the pulse is widening and contracting in a predetermined way such that the waveform across the output pin is sinusoidal. If we add appropriate filter (LC circuit) across the output we will see sine wave on the oscilloscope. Now we have to generate 2 more such waves with 120 degree phase difference with each other.

Why the 3 phases are with 120 degree phase difference?

This is a fundamental question; the brilliant mind Nikola Tesla designed the 3 phase generator at the beginning of the era of electrifying the world.

The 3 phase generator can produce thrice the power than single phase generator and running a motor at 3-phase also produced more torque than single phase motor.

You may interested in: [Three Phase Inverter Circuit Diagram.](#)

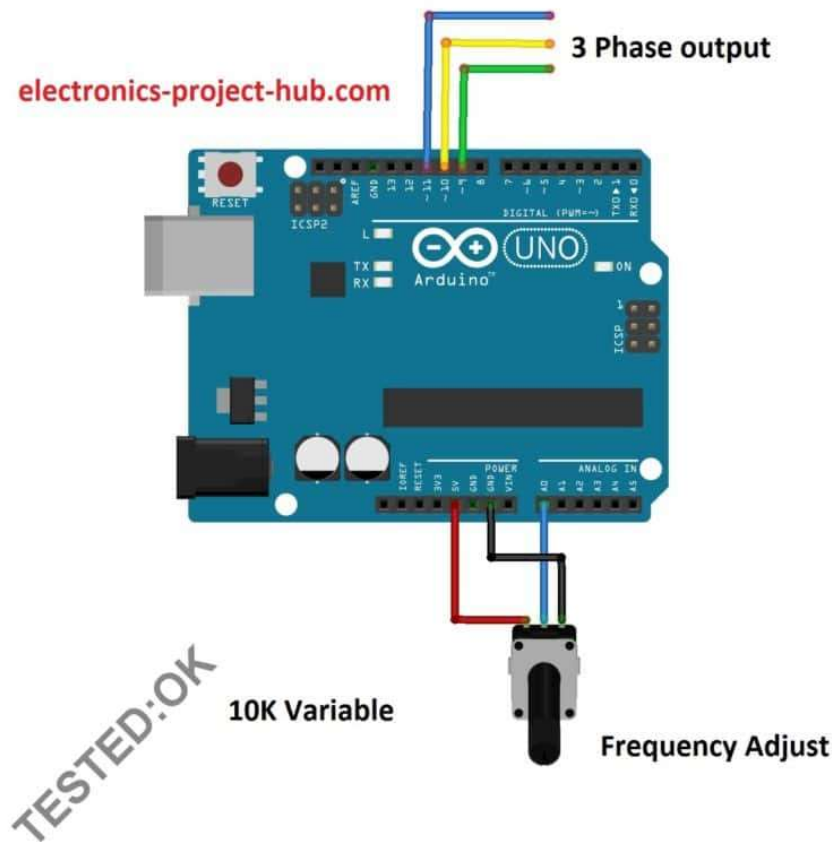
3-phase became a standard in industrial application because adding any more phases didn't help in increasing the efficiency or overall performance and neither justified the cost that involved in constructing in 6 or 12-phase for the output that we get.

The three coil winding inside the generator are placed with difference of 120 degree from each other for balanced rotational motion. $360 \text{ degree} / 3 \text{ phase winding} = 120 \text{ degree}$. Due to this the electrical output is with 120 degree phase difference.

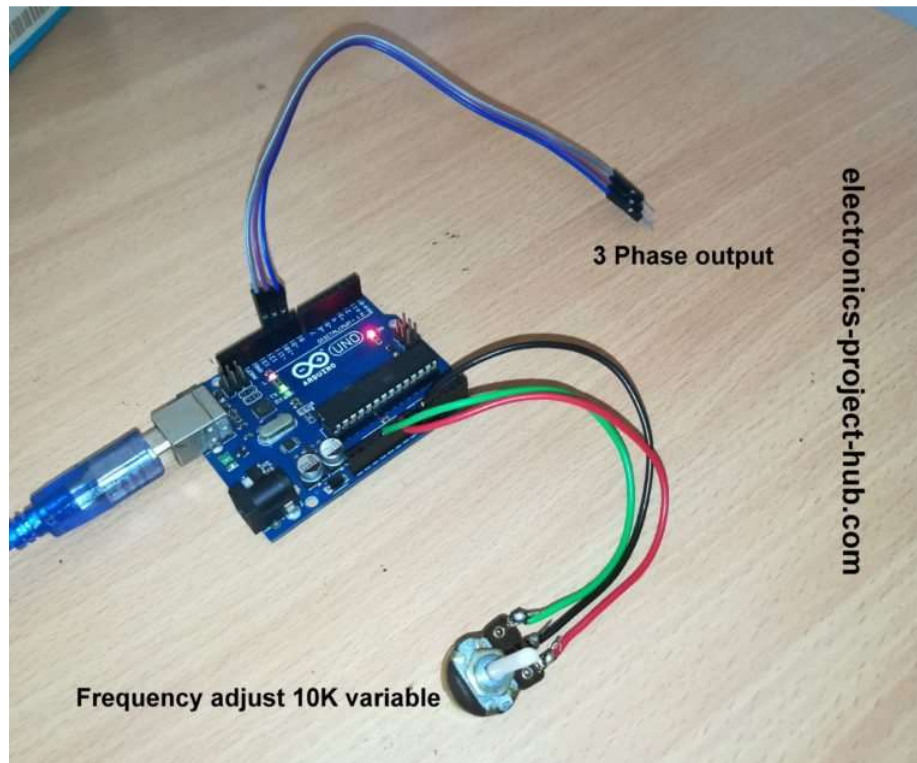
Now you know why the 3 phases are with 120 degree phase difference.

Hope you read the above theoretical explanation, now let's construct the circuit for generating 3-phase sine wave.

Circuit diagram:



3 Phase Sine Wave Generator Circuit



3 Phase Sine Wave Generator Circuit – Arduino

The circuit is very simple; it consists of an Arduino board and a 10k potentiometer for adjusting the frequency of 3-phase output.

The pin #9, #10 and #11 are the PWM pins have the capability to produce analog voltage level as described before; it produces pulse at 490 Hz by default. These three pins are tuned by the code to produce 3-phase sine wave.

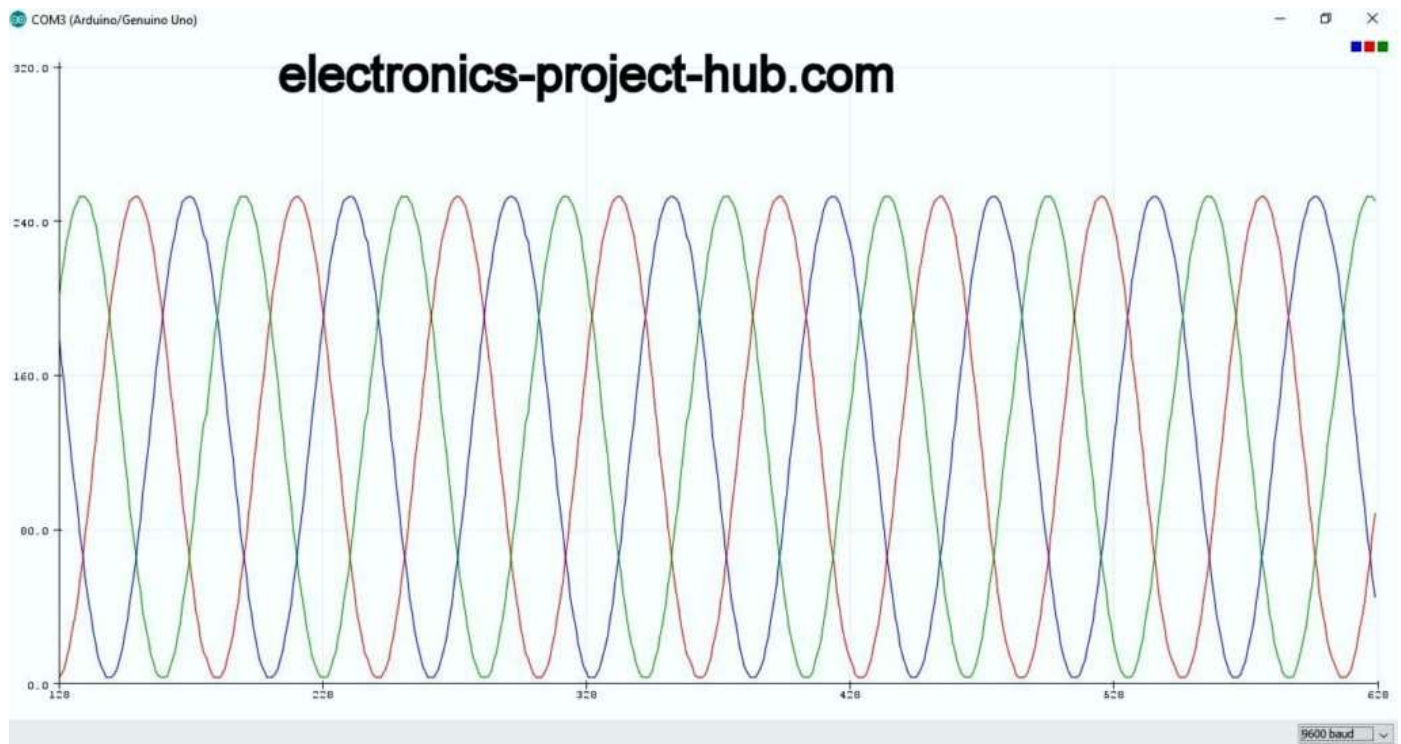
Program code: Verified

```
//-----www<electronics-project-hub>com-----//
#include <math.h>
int Output1 = 11;
int Output2 = 10;
int Output3 = 9;
int potVal = 0;
float A = 0;
float B = 0.104;
int Freq_IN = A0;
int var1 = 0;
int var2 = 0;
int var3 = 0;
int var4 = 0;
int var5 = 0;
int var6 = 0;
float Phase1 = 2 * PI / 3;
float Phase2 = 4 * PI / 3;
```

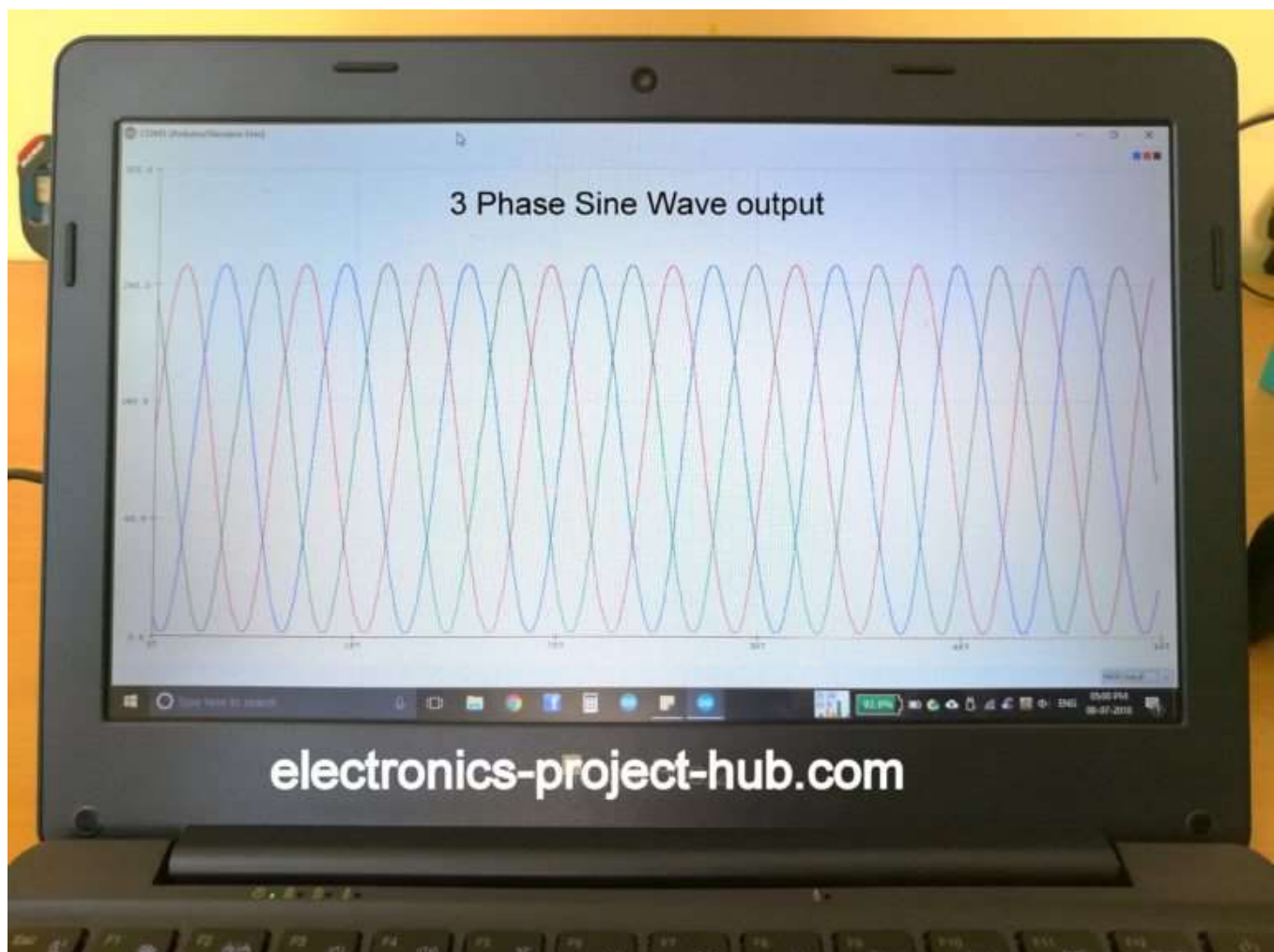
```
float Phase3 = 2 * PI;
boolean toggle = true; // true = Enabling Serial Plotter Output
void setup()
{
    Serial.begin(9600);
    pinMode(Output1, OUTPUT);
    pinMode(Output2, OUTPUT);
    pinMode(Output3, OUTPUT);
    pinMode(Freq_IN, INPUT);
}
void loop()
{
    A += B;
    analogWrite(Output1, var1);
    analogWrite(Output2, var2);
    analogWrite(Output3, var3);
    if (toggle == true)
    {
        Serial.print(var1);
        Serial.print(" ");
        Serial.print(var2);
        Serial.print(" ");
        Serial.println(var3);
    }
    var4 = 126 * sin(A + Phase1);
    var1 = var4 + 128;
    var5 = 126 * sin(A + Phase2);
    var2 = var5 + 128;
    var6 = 126 * sin(A + Phase3);
    var3 = var6 + 128;
    if (A >= 2 * PI)
    {
        A = 0;
    }
    potVal = analogRead(Freq_IN);
    delay(potVal);
}
//-----www<electronics-project-hub>com-----//
```

Will the above code produce 3-phase sine wave? let's check it out. The above code has inbuilt functionality to check the output waveform, you can check out by opening the serial plotter or by pressing **“Control + Shift + L”**.

Generated 3 Phase sine wave:



3 Phase sine wave



3 Phase sine waveform

On oscilloscope you will see the changing PWM width (which you saw on the GIF).

You may interested in: [Three Phase Inverter Circuit Diagram.](#)

Important Update:

Readers in the comment section are asking how to generate negative cycles of this 3 phase sine wave generator.

Answer:

Readers are asking this because they are measuring all the 3 phases with respect to ground (connecting the oscilloscope to ground and measuring individual phases) by doing so, the scope will shift the wave form vertically up and read 0 to 5V sine wave. If you measure each phases with respect to positive supply you will get down-shifted sine wave and read 0 to -5V sine wave.

But in reality, all the 3 phases are generating both negative and positive cycles with respect to each phases.

We hope this cleared readers question.

If you have any further question regarding this project, please comment below you can anticipate a guaranteed reply from us.

Blogthor



My nick name is **blogthor**, I am a professional electronics engineer specialized in Embedded System. I am a experienced programmer and electronics hardware developer. I am the founder of **this website**, I am also a hobbyist, DIYer and a constant learner. I love to solve your technical queries via comment section.

 Arduino, Electronics Basics



[report this ad](#)