



Development of Low-Cost High-Power Sinusoidal Function Generator for Driving a Single Tone Wave into Speaker using Arduino Uno and 2N3055

Adimas Wicaksana* and Sulanjari

*Mechanical Engineering, Faculty of Engineering, Universitas Pamulang
 Jl. Surya Kencana No. 1 Pamulang, Tangerang Selatan, Banten 15417, Indonesia*

E-mail: *dosen01678@unpam.ac.id

Masuk :21 Agustus 2018

Direvisi :26 September 2018

Disetujui :7 Januari 2019

Abstract: Function generator is one of the most important equipment in electronics testing. However its cost can be quite prohibitive. In this work we describe how to transform a square wave generated by Arduino into a sine wave by using filter/amplifier based on 2N3055, thus enabling low-cost high-power function generator. The electronics circuit was assembled using AWG22 wire and breadboard. Said function generator will be shown to be capable of producing sine wave with frequency between 500Hz-10KHz. With proper tuning it is capable of achieving up to 0.6% accuracy within the desired value. It costs less than IDR 150000 and capable of delivering 100W of power.

Keywords: Arduino, function generator, 2N3055

Abstrak: Function generator adalah salah satu alat paling penting untuk pengujian elektronik. Sayangnya biaya pembelian alat tersebut cukup mahal. Pada penelitian ini kami akan mempelajari cara mengubah gelombang kotak yang dihasilkan oleh Arduino menjadi gelombang sinus dengan menggunakan filter/amplifier berbasis 2N3055 sehingga dapat menghasilkan function generator berdaya tinggi dengan biaya rendah. Function generator tersebut mampu menghasilkan gelombang dengan frekuensi antara 500Hz-10kHz. Dengan prosedur penyeteman yang benar function generator ini dapat menghasilkan akurasi sampai dengan 0,6% dari angka yang ingin dicapai.

Kata kunci: Arduino, function generator, 2N3055

INTRODUCTION

Function generator plays a crucial role in electronics laboratory. It can be used to test the performance of an amplifier or measure the value of an unknown capacitor/inductor. The price of a commercial function generator, however, can be quite expensive. It can cost well above IDR 1000000 for a single function generator. While there are a single board solutions that cost less sometime one can't spare money to justify the purchase of an equipment whose sole function is to generate waveform.

On the other hand, an Arduino is a general purpose tool. A typical Arduino Uno costs around IDR 50000. After adding all the passive (inductor, capacitors) and active (2N3055) components the typical cost of equipment described in this paper is still lower than IDR 150000. This means that it is still quite competitive with respect to other single board solution that can be found online whose price ranges from IDR 75000-IDR200000. On top of that it can be reused for different purpose such as data acquisition[1] and speed control system [2].

THEORY

The problem with Arduino is that it is only capable of producing digital output which means that it is unable to produce a sinusoidal output. Nevertheless since Arduino can produce square wave and a square wave can be represented as summation of sinusoidal waves[3] one can generate a sinusoidal wave simply by filtering the higher harmonics. The summation can be described in the form of a picture as shown in Figure 1.

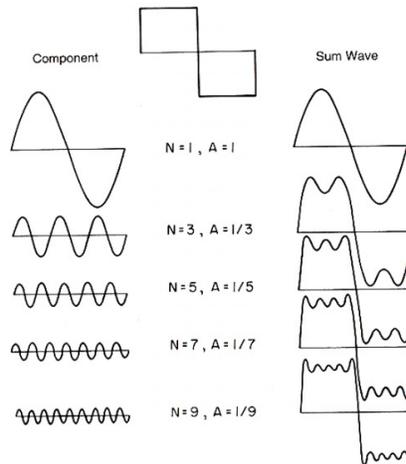


Figure 1. Square Wave as A Summation of Sine Wave [3]

To achieve this an L-C filter can be utilized in conjunction with 2N3055. Besides acting as a filter this particular circuit also serves as a class C power amplifier that is suitable for sine wave generator[4]. The amplifier is necessary because the original square wave signal is not quite strong enough to drive the speaker. Conceptually the function generator can be shown as in figure 2 below:

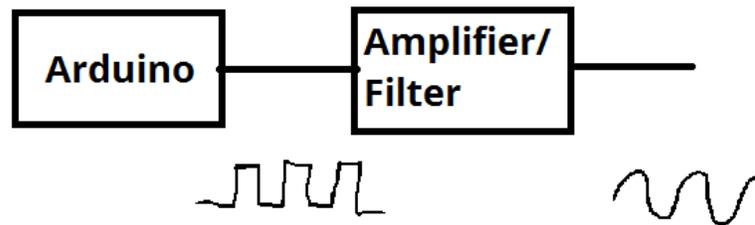


Figure 2. Conceptual Block Diagram

IC 2N3055 is chosen for its ability to drive relatively higher current[5]. This is also the reason why it is quite popular for high power application such as electricity generator[6] and high voltage DC power supply[7]. It is also capable of operating at higher power (>100W)[8].

METHODS

Arduino was first programmed using Arduino IDE (Integrated Development Environment) version 1.8.5 with a simple program that generates square wave. The program consists of 4 steps:

1. Drive the output pin into high(5V) state
2. Delay for $T/2$ seconds using delayMicrosecond function where T is the desired period of the waveform
3. Drive the output pin into low(0V) state
4. Delay for $T/2$ seconds using delayMicrosecond function and go back to step 1

These steps can be translated into C program shown in figure 3.

```

sketch_mar16a $
int outPin = 8;           // digital pin 8

void setup() {
  pinMode(outPin, OUTPUT); // sets the digital pin as output
}

void loop() {
  digitalWrite(outPin, HIGH); // sets the pin on
  delayMicroseconds(500);     // pauses for 50 microseconds
  digitalWrite(outPin, LOW);  // sets the pin off
  delayMicroseconds(500);     // pauses for 50 microseconds
}

```

Figure 3. Arduino Program Used to Generate Square Wave

Since the transition from state 0 to state 1 and vice versa would be abrupt the generated waveform would be a square. The wave would need to be filtered using L-C filter inside class C power amplifier. Below are the schematics:

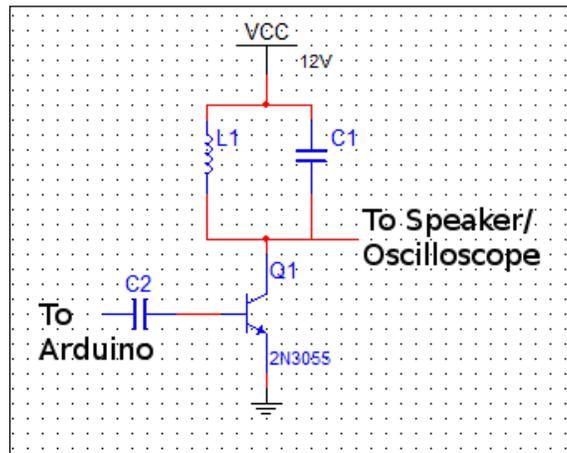


Figure 4. Schematics of the Filter/Amplifier

If gain control is deemed necessary a potentiometer can be added between the base and ground. An L-C bandpass filter has a peak frequency of [9]

$$f = \frac{1}{2\pi\sqrt{LC}} \quad (1)$$

Since the value of capacitor tends to be very low (pf-nf) and the required frequency is relatively low (between 500Hz and 10KHz) the value of inductor is selected to be as high as possible, in this case 0,25H. The value of capacitor vary from 1nf to 300nf depending on the desired frequency. A bypass capacitor is placed between the base and the emitter to act as a first stage low pass filter. The value is between 100pf and 10uf depending on the required frequency. Below are the values L1,C1, and C2 for each frequency tested:

Table 1. Design Parameter for Each Frequency Tested

Frequency (Hz)	L1(H)	C1(nF)	C2 (nF)	Arduino Delay (μ s)
581	0.25	300	1000	860
711	0.25	200	700	702
1000	0.25	100	300	500
10000	0.25	1	0.1	50

The electronic circuit described in figure above was then placed onto a breadboard. All connection in the breadboard was made using 22AWG cable. End connection to Arduino was made using standard 22AWG cable using as well. The other end was connected to either speaker or oscilloscope using alligator clip based on which test was performed.

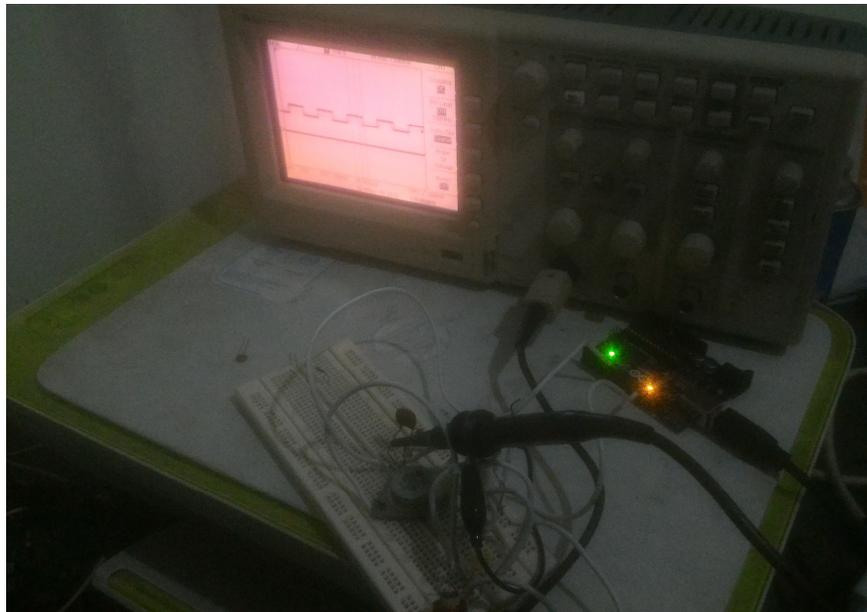
In this study two kinds of tests were performed to judge the performance of the function generator. In the first test oscilloscope was used to observe generated wave visually. In the second test the function generator was connected to a speaker whose output was then captured using microphone. The audio output can then be analyzed by a computer using Spectralab FFT software.

The followings are the Bill of Materials of all the components used in producing the square wave

Table 2. Bill of Material of Components Required

Components	Cost
Arduino	IDR 50,000
Inductor 0.25H	IDR 55,000
2N3055	IDR 10,000
Others(Capacitor, Breadboard, Wires)	IDR 20,000
Total	IDR 140,000

RESULT AND DISCUSSION

**Figure 5.** Square Wave Output from Arduino @10KHz

From Figure 5 we can see that the output generated from Arduino is a square wave and needs to be filtered further.

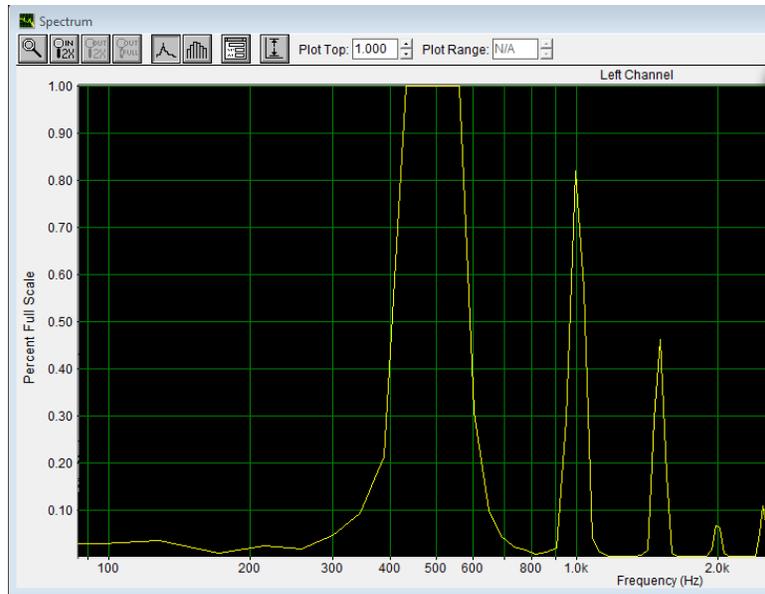


Figure 6. Square Wave Output from Arduino @500Hz

Figure 6 shows that SpectraLab FFT also shows higher harmonics at multiples of fundamental frequency, which is expected.

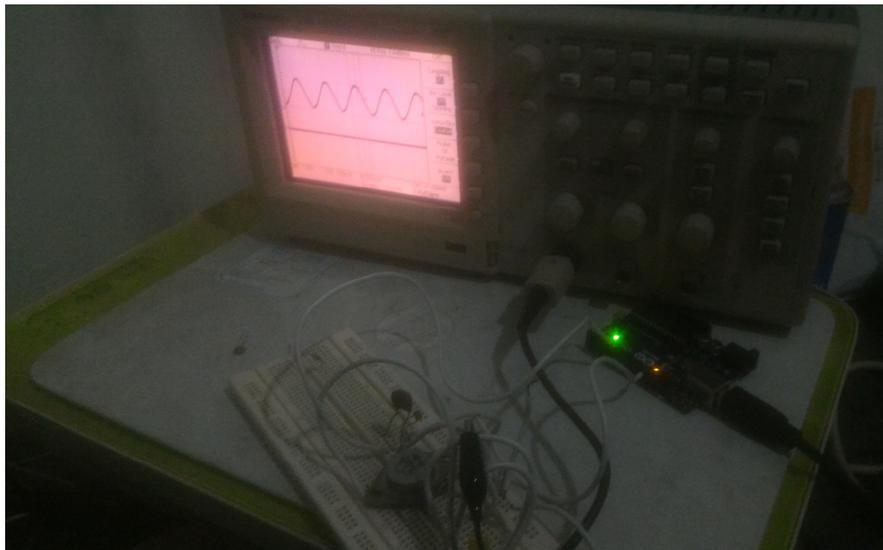


Figure 7. Sine Wave Output from Filter/Amplifier @10KHz

After filtering the output generated is a sine wave. Also notice the higher voltage due to the amplifier.

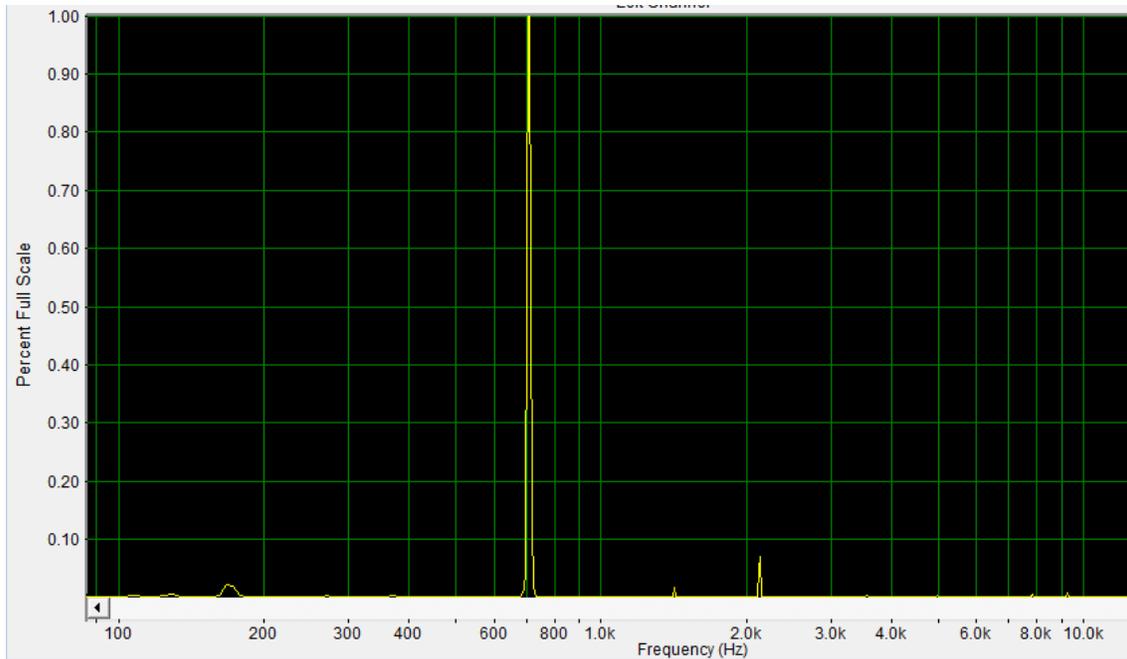


Figure 8. Spectrum Analysis after Filtering @700Hz

Figure 8 shows that output from SpectraLab FFT also shows significantly lower magnitude of higher harmonics.

Another observation noted was that the peak frequency detected by SpectraLab FFT is not exactly the same as the one programmed into Arduino. The frequency measured is typically around 1-3% lower than what is programmed. This is because there are certain overhead unaccounted for like the time required to set-up all the registers and the time required to jump inside the loop.

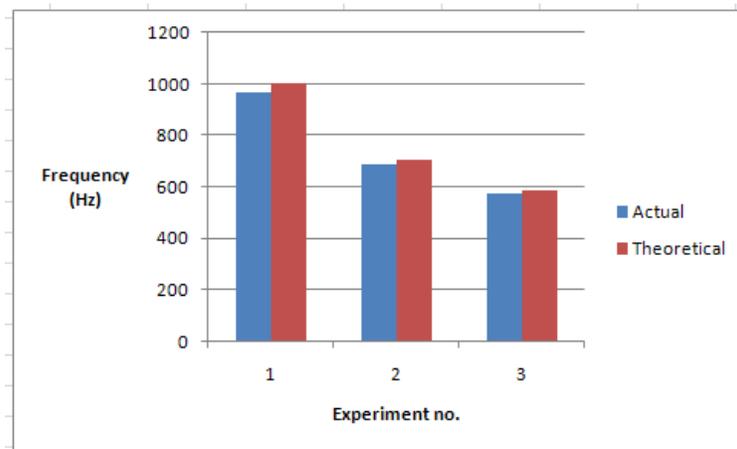


Figure 9. Lower Actual Frequency as Compared to Theoretical Ones

To compensate for this one can simply lower down the programmed delay so that the produced wave has the exact same frequency as designed. Figure 10 shows that we can achieve 0.6% accuracy for 1KHz frequency by changing Arduino delay value from 500 to 491.

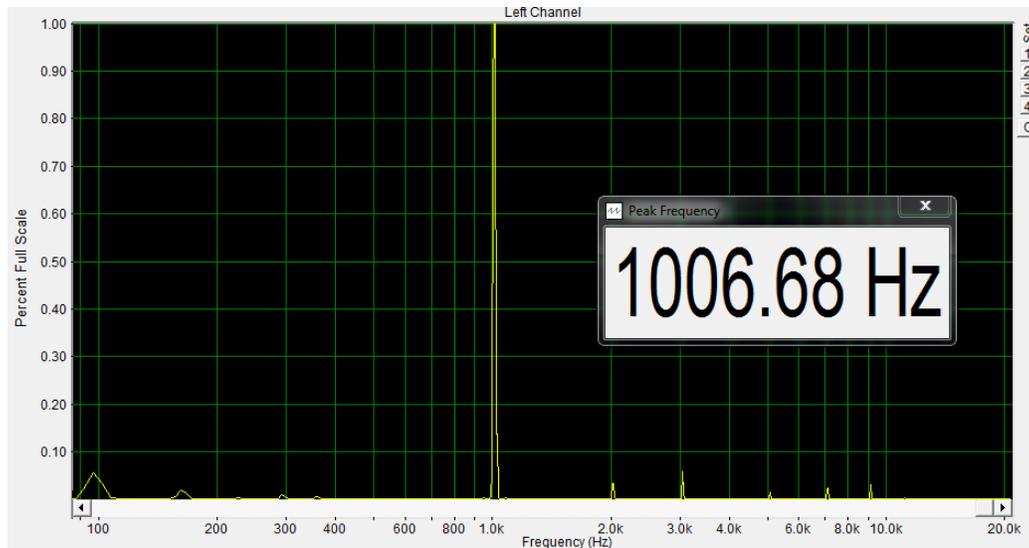


Figure 10. Accuracy of 0.6% Achieved @ 1KHz

CONCLUSIONS

A sine wave generator has been developed using Arduino and 2N3055-based class C Power Amplifier. The total cost required to produce this generator is less than IDR 140,000 and it is capable of driving around 100W of power. This function generator has been proven capable of producing output between 500Hz and 10KHz. While there can be discrepancy between the designed frequency and the actual frequency rendered this issue can be corrected simply by programming the Arduino at lower frequency.

REFERENCE

- [1] Yohan, Astuti F and Wicaksana A 2018 *J. Chimica et Nat. Acta.* **6**(3) 111
- [2] Rai N and Rai B 2013 *Int. J. of Engineering Trends and Technology* **4**(2) 137
- [3] Berg R E and Stork D G 2005 *The Physics of Sound* (San Fransisco:Pearson)
- [4] Smith J R 1997 *Modern Communication Circuits* (Singapore:McGraw-Hill)
- [5] Ellis J N and Osadchy V S 2001 *IEEE Transactions on Electron Devices* **48**(11) 2477
- [6] Maysha I, Trisno B and Hasbullah 2013 *Electrans* **12**(2) 89
- [7] Deb R and Sarma D K 2016 *Bulletin of Phys. Proj.* **1** 29
- [8] On Semiconductor 2005 *2N3055(NPN), MJ2955(PNP) Complementary Silicon Power Transistors.* (Arizona:SCILLC).
- [9] Odyniec M 2002 *RF and Microwave Oscillator Design* (Boston:Artech House)
- [10] Arduino LLC 2018 *Arduino Language Reference* <https://www.arduino.cc/reference/en/>