

Timing Circuit: NE555 Waveform Generator

ID: 1362467

Abstract

This paper is intended to provide the simulation of two applications using NE555 integrated circuit (IC) which is based on Empirical Modelling (EM). The two applications are mainly depended on the configuration of the circuit's components, so called 'Astable Mode' and 'Monostable Mode'. These modes are generally used in a variety of timing circuit. The scenario of this implementation can be simply taught as giving inputs to calculates outputs both in the form of number and graph. This signifies that the time or frequency of the NE555 IC will be able to calculate if components' values are known or vice versa. Thus, this implementation is aimed to educational purpose.

1 Introduction

The NE555 is an integrated circuit (IC) which is widely used to manipulate the timing or waveform generation circuits. It was introduced by the electronics manufacturer, named Signetics, in 1971. Although it exists since 1971, it still remains popular because of its low price, stability and ease of use for implementing a timing application.

Although there are many light software or online gadgets to compute NE555's equation, most of them are unhandy, difficult to analyse data, one-way calculation or require enough user's knowledge. Apparently, NE555 and its configuration are good example devices that demonstrate observables, dependencies, and agents. This will be explained in 'Em Concept' section.

Hence, in this project, the main objective is to create a user-friendly application for modelling the signal output of NE555 for two configurations which are astable and monostable mode based on EM's perspective. It provides connection of the circuit components in which how each component is wired to others, dynamically graphing and text results.

The structure of this paper is as follows: The preliminary description of the NE555 will be introduced providing general background knowledge. Secondly, the model and feature of the application will be listed and explained. Thirdly, the principles and concepts of EM will be justified consisting observables, dependencies, and agents. After that, the advantages and disadvantages are discussed which is followed by evaluation and conclusion.

2 Preliminary Description

2.1 Configuration Modes

As mentioned earlier, the output of NE555, usually in waveform, can be obtained by constructing suitable components in a correct position of a circuit and selecting the desired values of components. Thus, this rises two well-known configurations in which the NE555 can be operated either in a monostable (one-shot) mode or in an astable (free-running) mode.

The behaviour of NE555's waveform, for all modes, can be generally adjusted by changing the values of associated resistor(s) or capacitor(s). The further details on astable and monostable mode are shown in Appendix 1 which contains equations and circuit connections.

3 Model and Feature

3.1 Model

The structure of the application has two separately different interfaces which do not overlap to each others, referred to Appendix 2. Hence, one interface is to simulate the astable mode and another interface is to simulate the monostable mode. Nevertheless, it can be switched back and forth freely. As two modes contains similar functions and structure, each interface consists of the circuit diagram, input data, and the simulated output results which are displayed as texts and graph. The detail please refer to Appendix 4

The purpose of circuit diagram is to demonstrate the connection between components and their values. Please note that component values can be as either input or output depending on the given data. They will act like inputs only if the component values are given from user and they will act like the outputs only if timing is given i.e. reverse calculation. Moreover, the circuit diagram provides brief components' information when mouse cursor hovers them.

The layout of input data is where inputs are given to analyse. The inputs can be classified into two types. The first type is where the values for each component are known. This type is used when the timing of NE555 is interested in particular values of component. The second type is to calculate the components' values where the timing is known.

The simulation part is where the graph is plotted according to the output results after inputs are correctly set. The graph is aimed to provide the better look of results which gives the clear illustration of waveform. These contents will be explained more in 'EM Concept' section.

3.2 Feature

The features of this application are as follows:

- Each mode contains couple calculations i.e. giving all component values to obtain timing or setting timing to obtain component values.
- All inputs are firstly checked their formats to ensure the calculation is correct i.e. they must be number format only. Otherwise, '#invalid' phase will display.
- Once inputs are correctly set, each of them can be slightly adjusted via slider bar to investigate the dynamic results. Each input can be decreased or increased in the range of -100 to 100.
- Results can be approximated by enabling approximation's tick box. This will give clean and better view of results.
- Brief information for each component (NE555, resistor, and capacitor) will be displayed when mouse cursor hovers it. This will provide the effect to the output of each component.

4 EM Concept

The EM concept is referred to the development of construal for the behaviours of the NE555's waveform between two modes. The EM concept consists of observables, dependencies, and agents which will be explained.

4.1 Observables

There are many observables used to illustrate the structure of this application. Within this application, some of these are the circuit diagrams providing the connection and information of the circuit, the graph to plot and display output results as text including their colours. The visualisation of circuit diagram contains much more observables that demonstrate

the connection between each component in the real world. The visualisation of graph is used to reveal the behaviours of the output waveform of NE555 more clearly and understandably. Moreover, the observables that represent precise results in number format are displayed as text form. Therefore, these observables behave according to other observables' behaviours or to the given inputs which mean dependencies.

4.2 Dependencies

The dependencies are relations between those observables that interact each other in one way or another. The graph is apparently depended on the results which is calculated from (depended on) the given inputs from the user. This signifies that the length representing high time and low time is depended on the value of results. Moreover, the graph is plotted either in static mode or in dynamic mode. This means that if the inputs are given in text boxes, the graph is in static mode (fixed value). If the slider bar is, however, used to control the individual input value, the graph will be in dynamic mode (varying over time). Such approaches give the new opportunity to learn the context and knowledge how does the waveform of NE555 look like or change in the real world from this application.

Further more on displaying results, the results can be displayed either precise values (more than ten decimals) or approximated values by ticking the approximation box (round up to two decimals). This signifies that the results are not only depended on the given input values but also depended on the approximation.

Moreover, the information of components are movably displayed, following mouse cursor, whenever mouse cursor hovers any of them. This implies that their positions are depended on the mouse position in x- and y-axis.

4.3 Agents

Some agents, which are created and used in this application, are as follows: There is an agent used for checking the typo. This agent will check the format of inputs i.e. it must be a number with one dot to represent decimal only. If the typo is occurred by any means, the agent will restrain that given input, and will display as "#invalid" on screen instead of given value, and will stop the calculation and graph as well. This action will prevent uncontrollable problem during calculation by given wrong input.

As the mouse's actions, such as movement, pressing, etc., are conditionally checked, the trigger actions will execute at some stages if their conditions are

met. The examples are: whenever the mouse is pressed in the area of menu (astable mode or monostable mode), this action will trigger the user's selection which will change the interface. Moreover, the brief explanation will pop-up if mouse cursor hovers on components.

5 Advantages and Disadvantage

5.1 Advantages

This application is intended to develop and run in JS-Eden (master) rather than TKEden. The reasons why JS-Eden is chosen are as follows: In term of object's visualisation, TKEden required more than one notation to create and illustrate an object (using scout and donald) and has limitation of shape. However, JS-Eden provides more flexibility in drawing an object within one language, for example, polygon, arc, point, etc. Therefore, having this feature, the fancy shape can be easily created which improves visualisation.

As the JS-Eden is based on JavaScript's implementation, the JavaScript's embedded methods can be freely used to extend the potential of various functions or agents to response more dynamically. For example, it is not easy to obtain the value of sine function or rounding number in TKEden as they can be done easily in JS-Eden.

Moreover, triggered action are common EDEN construction and it is available in JS-Eden as well. By combining triggered action, the application can be implemented to become even more complex interactive behaviours i.e. to created an agent.

Although some features of implementation can be written in TKEden, the syntaxes might be complex and eventually lead to bugs.

Moreover, it is easier to create an application in Js-Eden language than to use other traditional language in term of code's structure and style. By using "is" or triggered actions, there is no need to create functions to recall and to update values any time during run time.

5.2 Disadvantages

This application requires the bigger size of Canvas HTML5 to display the whole program which can be done by either reassigning value of Canvas or manually adjusting its size by mouse. Thus, the former method is used in this case i.e. the code itself contains the new size of Canvas. However, the size of the Canvas is not changed according to the values after the first code's execution. Although the size of Canvas is bigger after the second code's execution,

the display area is still the same. This means that some areas are cropped. This is an annoying situation as a user must manually drag at its corner to adjust its size every time after code's execution. This might be a problem inside JavaScript's interpretation. The assumption is that the code is too long and needs some times to execute which contribute to wrong behaviour of Canvas during modifying its size.

There is no debugging tool available in JS-Eden that allows programmer to investigate the behaviour of a program from line to line code. Therefore, debugging process might be difficult. The easiest way, however, is to check the program's behaviour frequently while developing a program. It is recommended to use "Symbol List" or "Symbol Lookup Table" to check the individual observable as well.

6 Evaluation

In order to determine the efficiency of this application, the evaluation is applied. This can be done by evaluating the output data whether the output data is satisfied from the theoretical conception or not. If not, how much the error is produced.

The application is evaluated by comparing the results obtained by itself and freeware NE555 generator called "555 timer component selection" from DOCTRONICS website[1]. The detail please refer to Appendix 5.

After evaluation, this application is concluded to be accurate enough to provide good results.

7 Conclusion

The visualisation of this application gives excellent learning tool by taking the advantages of observables, dependencies and agents. As this NE555's simulator has rich observables, dependencies and agents, it is a good example to investigate. By comparing a traditional simulation, this application has unique quality. Learners, for example, can observe the effect of component's characteristic to the output waveform. Moreover, this application or basic idea can be further applied to itself or to other similar area.

References

- [1] Dr W D Phillips. (2008). *555 Timer*. Available: <http://www.doctrionics.co.uk/555.htm>. Last accessed 28th Jan 2014.
- [2] embeddedfortheevil. (2010). *555 Timer IC-Introduction*. Available: <http://embedded-fortheevil.wordpress.com/2010/10/30/555->

timer-ic-introduction/. Last accessed 28th Jan 2014.

[3] Glen A. Williamson . (2011). *555 Animation*. Available: http://www.williamson-labs.com/480_555.htm. Last accessed 28th Jan 2014.

[4] Amandaghassaei. *555 Timer*. Available: <http://www.instructables.com/id/555-Timer/>. Last accessed 28th Jan 2014.

Appendix 1

Monostable Mode

The output of monostable mode is to act like one pulse signal generator. The output is produced only when the NE555 is received a signal at the trigger pin (pin 2) and its length is depended on the RC network which is composed of a resistor (R) and a capacitor (C). Therefore, the output signal can be shortened and lengthened to the need of application by changing the value of these two components. The time for NE555 to generate the output signal is given by

$$t = RC \ln(3) \approx 1.1RC$$

t is the time taken for NE555 to rise the output in second which is the multiplication of R, C, and natural logarithm of three. This term, however, can be replaced by approximated value to be more convenient. The circuit of monostable mode is shown in diagram 1.

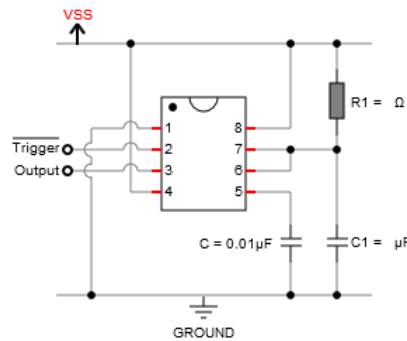


Diagram 1: Circuit for Monostable Mode

Astable Mode

For the astable mode, the output is considered as continuous rectangular waveform pulse signal whose its frequency and duty cycle are mainly depended on two external resistors (R) and one capacitor (C). As one duty cycle contains high and low output signal which determine its frequency, the equations can be given by.

$$T_{High} = \ln(2)(R_1 + R_2)C \approx 0.7(R_1 + R_2)C \quad T_{Low} = \ln(2)(R_2)C \approx 0.7(R_2)C$$

$$T_{Total} = \ln(2)(R_1 + 2R_2)C \approx 0.7(R_1 + 2R_2)C \quad f = \frac{1}{\ln(2)(R_1 + 2R_2)C} \approx \frac{1.45}{(R_1 + 2R_2)C} \quad Duty\ Cycle = \frac{(R_1 + R_2)}{(R_1 + 2R_2)} * 100$$

It can be seen from above equations that the output signal can be freely adjusted by varying the component values. The circuit of astable mode is shown in diagram 2.

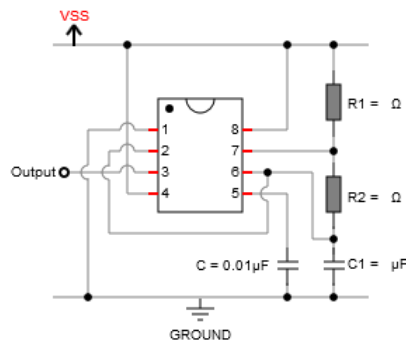
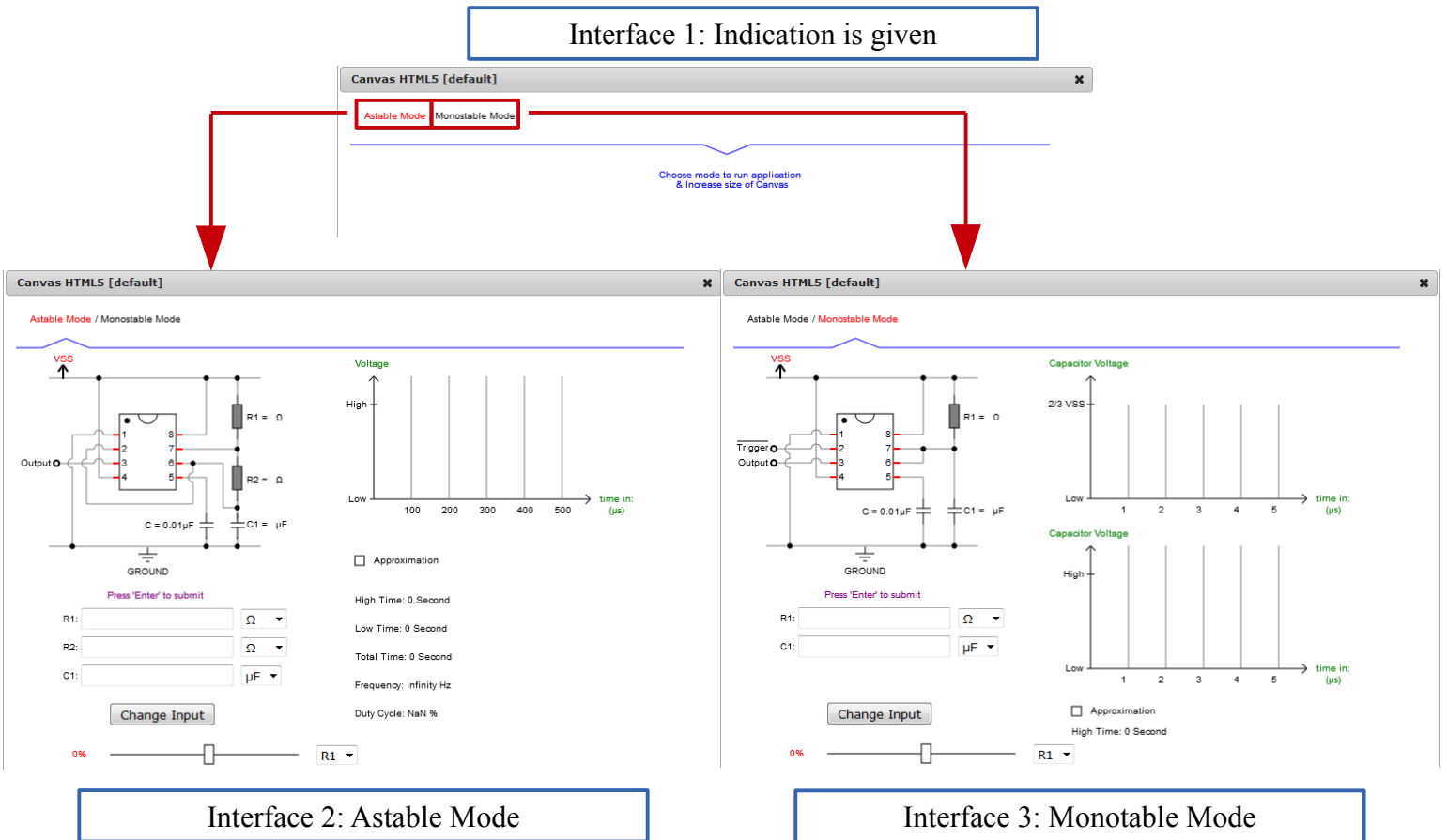


Diagram 2: Circuit for Astable Mode

On the other hands, the desired frequency or timing output signal, for both modes, can be determined by means of rearranging the equations to get R and C.

Appendix 2 - Interfaces

After submitting the whole codes into JS-Eden, the first interface will be displayed. The first interface is used to overcome the problem occurred during first time codes' submission as it does not display neither astable mode nor monostable mode unless user clicks either of them. Therefore, the first interface gives indication. If the astable mode is clicked, the interface is changed to interface of astable mode. If the monostable mode is clicked, the interface is changed to interface of monostable mode. Please note that it can be changed freely at any time back and forth..



Appendix 3 - Sample Calculation

To calculate the results, the sufficient inputs must be given: given component values of R1, R2, and C1 to calculate the high time, low time, and duty cycle or given frequency, duty cycle, and C1 to calculate R1, R2.

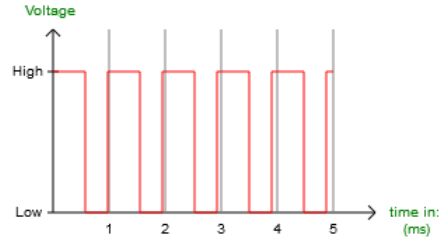
suppose we want to find the frequency and we have $R1 = 288 \Omega$, $R2 = 567 \Omega$ and $C1 = 10^{-6} \mu\text{F}$, therefore

$$T_{High} = \ln(2)(288+567)10^{-6} \approx 0.000593$$

$$T_{Low} = \ln(2)(567)10^{-6} \approx 0.000393$$

$$T_{Total} = \ln(2)(288+2(567))10^{-6} \approx 0.000986$$

$$Duty\ Cycle = \frac{(288+567)}{(288+(2*567))} * 100 \approx 60.126$$



Approximation

High Time: 0.000593 Second

Low Time: 0.000393 Second

Total Time: 0.000986 Second

Frequency: 1010 Hz

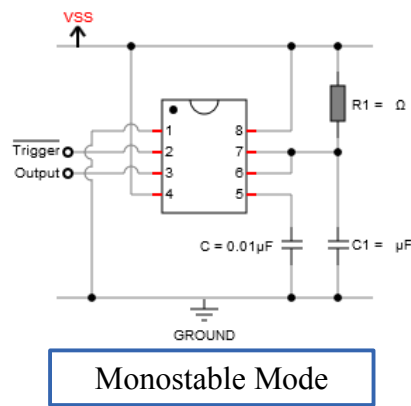
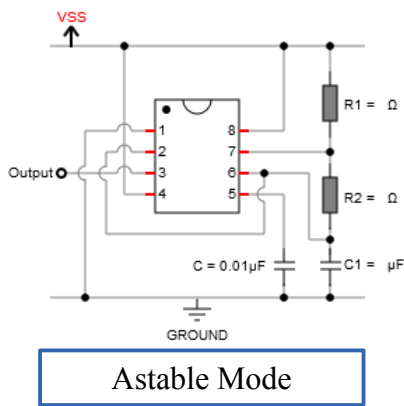
Duty Cycle: 60.1 %

Appendix 4 – Modeling Structure

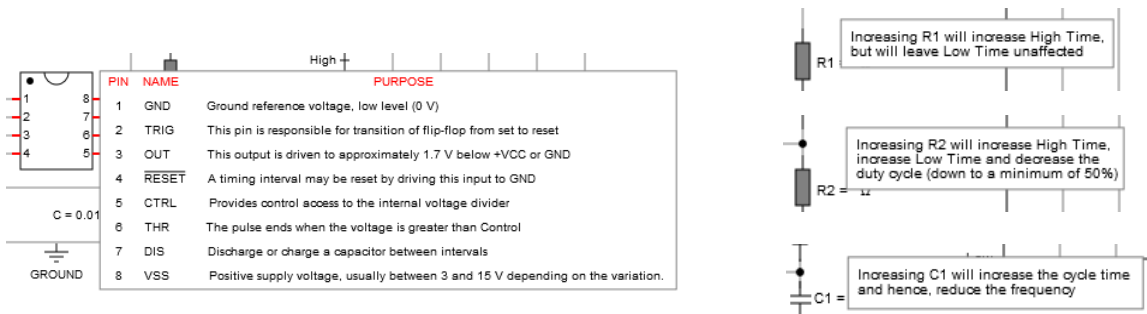
Circuit Diagram

The first structure of this application is circuit diagram. The diagrams show the circuit of Astable mode and Monostable mode. For astable mode, it can be seen that there are three components needed which are R1, R2, and C1. Please note that the value of C is fixed and connected to ground. Apart from C, these components can be varied in which the waveform of the NE555 is changed or affected every time any one of them is changed. Please note that the VSS is the representation as voltage source that supplies voltage to the circuit which range between 3 to 15 Volts in the real world. It is nothing to do with this application but it is only for demonstration. The Output (pin 3) is the presentation as the output of the NE555, its behavior is shown in graph. This output pin is usually driven approximately 1.7 volts below VSS.

For monostable mode, it can be seen that there are only two components needed – R1 and C1 – and the connections between each component are quiet similar to the connection of astable mode. Addition to VSS and Output (pin3), the Trigger (pin2) is inverted signal meaning that if the signal of this Trigger pin is low, it will become high or vice versa. This mode provides the one-shot waveform. On the other words, the NE555 will produce waveform only when the Trigger pin is low. It will stop when the capacitor is charged to about 2/3 of the supply voltage. Please note that time taken for capacitor to charge is depended on R1.



Beside of connection's illustration, there is a feature to give brief information about components and this feature is called whenever the mouse cursor hovers them.



Input data

There are two types of input data for both modes – the first type is where all component values are known, the second type is where the timing is known. The diagram below shows the first type of both mode. It can be seen that each unit has metric prefix – k = 1000, m = 1000000, μ = 0.000001 (10^{-6}), n = 0.000000001 (10^{-9}), p = 0.000000000001 (10^{-12})

Press 'Enter' to submit

R1: Ω Ω kΩ mΩ

R2: Ω Ω kΩ mΩ

C1: μF μF nF pF

Type 1: Astable Mode

Press 'Enter' to submit

R1: Ω Ω kΩ mΩ

C1: μF μF nF pF

Type 1: Monostable Mode

The diagram below shows the second type.

Press 'Enter' to submit

Frequency: Hz Hz kHz

Duty Cycle: %

C1: μF μF nF pF

Type 2: Astable Mode

Press 'Enter' to submit

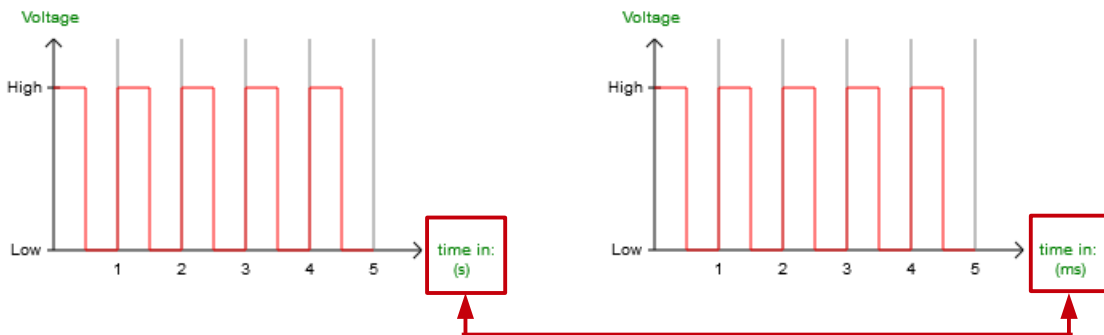
Time: s s ms

C1: μF μF nF pF

Type 2: Monostable Mode

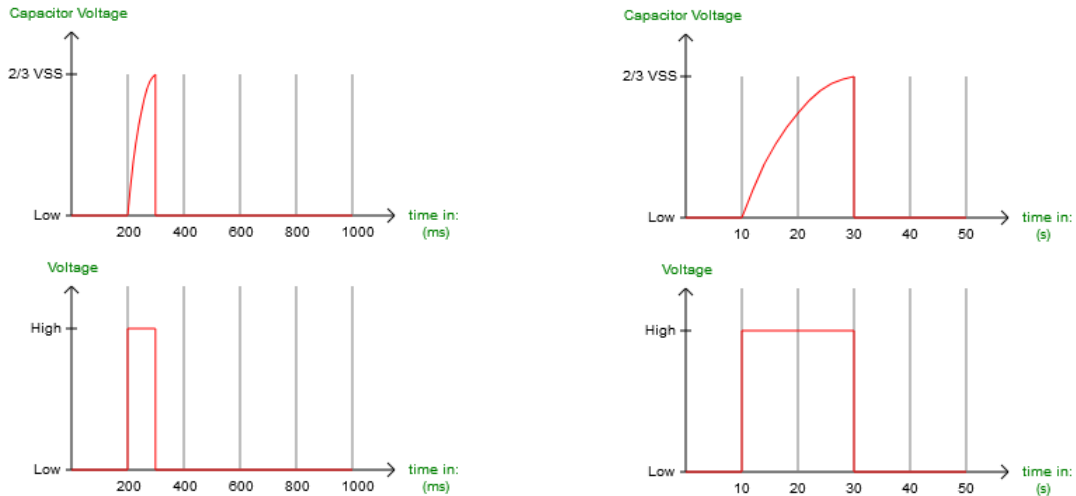
Graph

The graph of astable mode shows the waveform of NE555 in time domain. The y-axis represents voltage between high and low i.e. high voltage has current and low voltage has no current. The x-axis represents time in which its interval can be automatically changed depending on the results. Please look at its unit carefully - even the graph looks the same but the time's unit is different. This means that the waveform is different.



Graph: Astable Mode

There are two graphs of monostable mode providing the output waveform and capacitor charging voltage. The concept is pretty much the same. The y-axis represents voltage between high and low i.e. high voltage has current and low voltage has no current. The x-axis represents time in which its interval can be automatically changed depending on the results. The output waveform shows the waveform of NE555 in time domain and the capacitor charging voltage shows the voltage is being charge to capacitor.



Graph: Monostable Mode

Results

The results of both modes can be made very accuracy or approximation by clicking the box of approximation.

<input type="checkbox"/> Approximation	<input checked="" type="checkbox"/> Approximation
High Time: 0.5995723111843526 Second	High Time: 0.6 Second
Low Time: 0.39925277600252845 Second	Low Time: 0.399 Second
Total Time: 0.998825087186881 Second	Total Time: 0.999 Second
Frequency: 1.0011762948570184 Hz	Frequency: 1 Hz
Duty Cycle: 60.02775850104094 %	Duty Cycle: 60 %

Results: Astable Mode

<input type="checkbox"/> Approximation	<input checked="" type="checkbox"/> Approximation
High Time: 0.3185975637137518 Second	High Time: 0.319 Second

Results: Monostable Mode

Appendix 5 - Freeware Evaluation

The diagram below shows the comparison between freeware named “555 timer component selection” and application to evaluate the results. It is examined on astable mode having $R1 = 1\text{ k}\Omega$, $R2 = 100\text{ k}\Omega$ and $C1 = 10^{-6}\text{ }\mu\text{F}$ and on monostable mode having $R1 = 1200\text{ }\Omega$ and $C1 = 10^{-6}\text{ }\mu\text{F}$. It can be seen that the value of duty cycle and pulse (which is frequency) from both generators are similarly equal.

