

ELECTRONICS HUB

PROJECTS | TUTORIALS | COURSES | KITS

[HOME](#)[PROJECTS](#)[MINI PROJECTS](#)[FREE CIRCUITS](#)[TUTORIALS](#)[SYMBOLS](#)[DIY](#)[PROJECT KITS](#)[COURSES](#)[CONTACT US](#)

YOU ARE HERE: [HOME](#) / [COMBINATIONAL LOGIC CIRUCITS](#) / DIGITAL COMPARATOR AND MAGNITUDE COMPARATOR

Digital Comparator and Magnitude Comparator

JULY 7, 2015 BY [ADMINISTRATOR](#) — [7 COMMENTS](#)

Data comparison is needed in digital systems while performing arithmetic or logical operations. This comparison determines whether one number is greater than, equal, or less than the other number. A digital comparator is widely used in combinational system and is specially designed to compare the relative magnitudes of binary numbers.

These are also available in IC form with different bit comparing configurations such as 4-bit, 8-bit, etc. More than one comparator can also be connected in cascade arrangement to perform comparison of numbers of longer lengths. Whenever we want to compare the two binary numbers, first we have to compare the most significant bits.

If these MSBs are equal, then only we need to compare the next significant bits. But if the MSBs are not equal, then it would be clear that either A is greater than or less than B and the process of comparison ceases.

For example, the two 2-bit number are $A = A_1A_0$ and $B = B_1B_0$. If A_1 is not equal to B_1 , then it is clear that A is greater than B for $A_1 = 1$ & $B_1 = 0$ or else A is less than B for $A_0 = 0$ & $B_0 = 1$. At this stage the process of comparison ceases.

If the MSBs are equal, i.e., $A_1 = B_1$ only then we need to compare the next significant bits A_0 and B_0 and decide whether the number is greater than, less than or equal. So, the comparator produces three outputs as L, E and G corresponds to less than, equal and greater than comparisons.

Digital comparators can be of two types

Identity Comparator

Comparators that have only one output terminal and produces the output either low or high are identity comparators.

Magnitude Comparator

Comparators with three output terminals and checks for three conditions i.e greater than or less than or equal to is magnitude comparator.

[Back to top](#)

Table of Contents

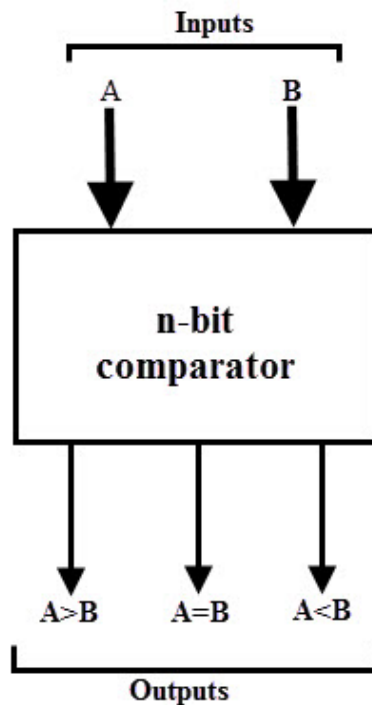


1. Digital Comparator
2. Single Bit Magnitude Comparator
3. 2-Bit Comparator
4. 4-Bit Comparator
5. 8-Bit Comparator
6. Applications of Comparators

Digital Comparator

A magnitude digital comparator is a combinational circuit that compares two digital or binary numbers (consider A and B) and determines their relative magnitudes in order to find out whether one number is equal, less than or greater than the other digital number.

Three binary variables are used to indicate the outcome of the comparison as $A > B$, $A < B$, or $A = B$. The below figure shows the block diagram of a n-bit comparator which compares the two numbers of n-bit length and generates their relation between themselves.



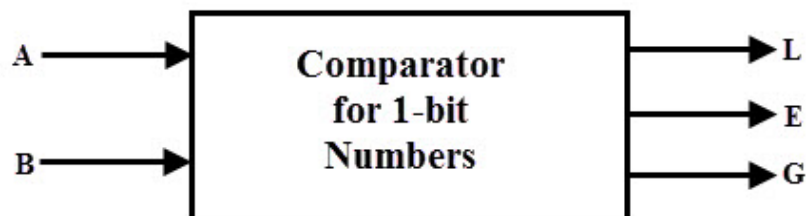
These comparators can compare 2-bit, 4-bit and 8-bit numbers depending on the application requirement. These are available in TTL as well as CMOS logic family ICs and some of these ICs include IC 7485 (4-bit comparator), IC 4585 (4-bit comparator in CMOS family) and IC 74AS885 (8-bit comparator).

[Back to top](#)

Single Bit Magnitude Comparator

A comparator used to compare two bits, i.e., two numbers each of single bit is called a single bit comparator. It consists of two inputs for allowing two single bit numbers and three outputs to generate less than, equal and greater than comparison outputs.

The figure below shows the block diagram of a single bit magnitude comparator. This comparator compares the two bits and produces one of the 3 outputs as L ($A < B$), E ($A = B$) and G ($A > B$).



The truth table for the single bit comparator is given below. When $A_0 B_0 = 00$ & 11 , both inputs are equal, therefore $A=B$ output will be high. When $A_0 B_0 = 01$, B is more than A and hence AB is active.

A_0	B_0	L	E	G
0	0	0	1	0
0	1	1	0	0
1	0	0	0	1
1	1	0	1	0

From the truth table logical expressions for each output can be expressed as

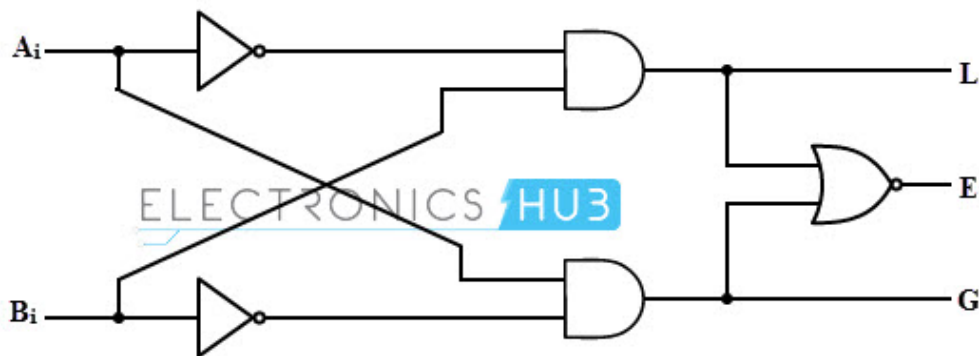
$$A_0 < B_0: L = \overline{A_0} B_0$$

$$A_0 = B_0: E = \overline{A_0} \overline{B_0} + A_0 B_0$$

$$A_0 > B_0: G = A_0 \overline{B_0}$$

It is to be noted that E can be realized as $\overline{(L + G)}$.

By using these Boolean expressions, we can implement a logic circuit for this comparator using two AND gates, one NOT gate and one Ex-NOR gate as shown in below figure. AND gates are used to find whether a binary digit is less than greater than another bit whereas Ex-NOR gate is used to find whether two binary numbers are equal or not.



In the figure, one AND gate has inputs of $A_0 (B_0)'$ and another has inputs $(A_0)' B_0$. Therefore, one AND gate output is 1 if $A_0 > B_0$ (i.e., $A_0 = 1$ and $B_0 = 0$) and is zero if $A_0 < B_0$ (i.e., $A_0 = 0$ and $B_0 = 1$). Similarly, other AND gate output is one if $A_0 < B_0$ (i.e., $A_0 = 0$ and $B_0 = 1$) and is zero if $A_0 > B_0$ (i.e., $A_0 = 1$ and $B_0 = 0$).

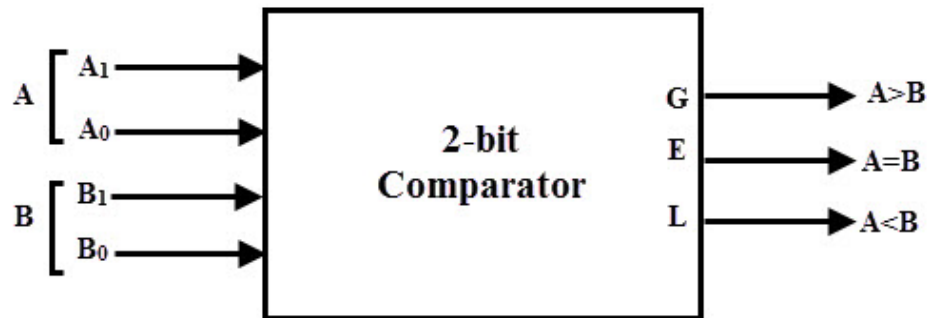
The Ex-NOR gate has inputs $A_0 B_0$, hence the output of the Ex-NOR gate will be 1 if $A_0 = B_0$ and the output will be 0 if A_0 is not equal to B_0 .

[Back to top](#)

2-Bit Comparator

A 2-bit comparator compares two binary numbers, each of two bits and produces their relation such as one number is equal or greater than or less than the other. The figure below shows the block diagram of a two-bit comparator which has four inputs and three outputs.

The first number A is designated as $A = A_1A_0$ and the second number is designated as $B = B_1B_0$. This comparator produces three outputs as G ($G = 1$ if $A > B$), E ($E = 1$, if $A = B$) and L ($L = 1$ if $A < B$).



The truth table of this comparator is shown below which depicting various input and output states.

Inputs				Outputs		
A_1	A_0	B_1	B_0	$A > B$	$A = B$	$A < B$
0	0	0	0	0	1	0
0	0	0	1	0	0	1
0	0	1	0	0	0	1
0	0	1	1	0	0	1
0	1	0	0	1	0	0
0	1	0	1	0	1	0
0	1	1	0	0	0	1
0	1	1	1	0	0	1
1	0	0	0	1	0	0
1	0	0	1	1	0	0
1	0	1	0	0	1	0
1	0	1	1	0	0	1
1	1	0	0	1	0	0
1	1	0	1	1	0	0
1	1	1	0	1	0	0
1	1	1	1	0	1	0

The k-map simplification for the above truth table is as follows.

		A>B						A=B			
A₁A₀	B₁B₀	00	01	11	10	A₁A₀	B₁B₀	00	01	11	10
00		0	0	0	0	00		1	0	0	0
01		1	0	0	0	01		0	1	0	0
11		1	1	0	1	11		0	0	1	0
10		1	1	0	0	10		0	0	0	1

From the above k-map simplification, each output can be expressed as

$$A > B: G = A_0 \overline{B_1} \overline{B_0} + A_1 \overline{B_1} + A_1 A_0 \overline{B_0}$$

$$A = B: E = \overline{A_1} \overline{A_0} \overline{B_1} \overline{B_0} + \overline{A_1} A_0 \overline{B_1} B_0 + A_1 A_0 B_1 B_0 + A_1 \overline{A_0} B_1 \overline{B_0}$$

$$= \overline{A_1} \overline{B_1} (\overline{A_0} \overline{B_0} + A_0 B_0) + A_1 B_1 (A_0 B_0 + \overline{A_0} \overline{B_0})$$

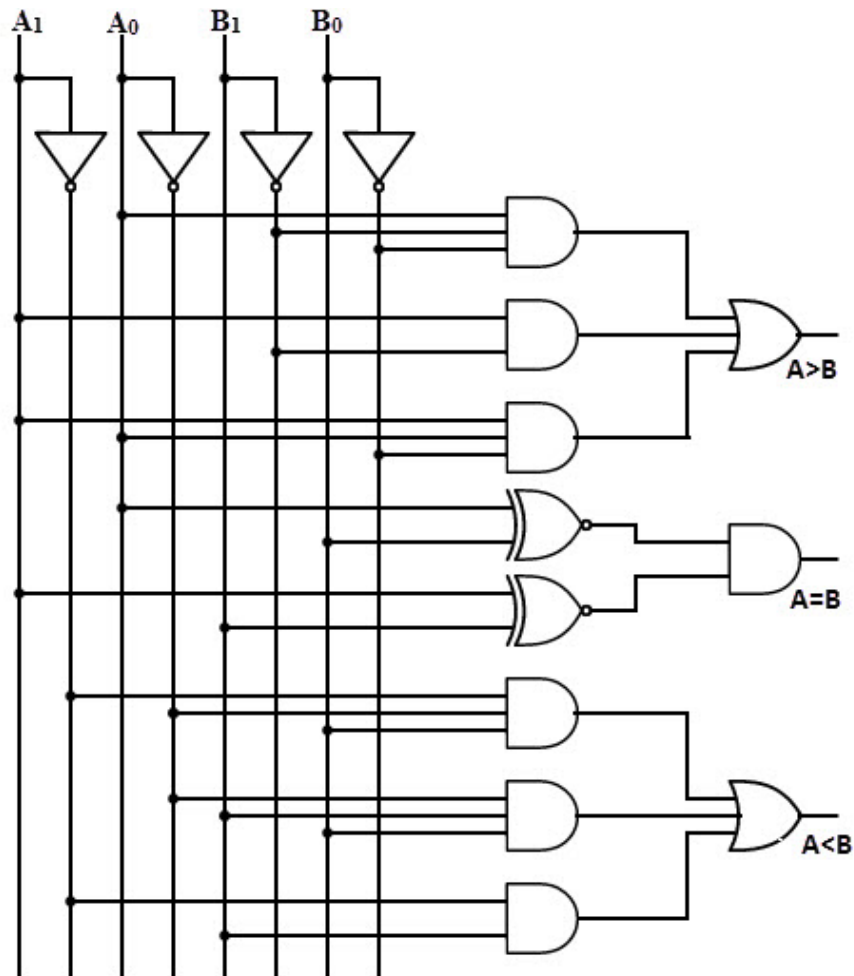
$$= (A_0 B_0 + \overline{A_0} \overline{B_0}) (A_1 B_1 + \overline{A_1} \overline{B_1})$$

$$= (A_0 \text{ Ex-NOR } B_0) (A_1 \text{ Ex-NOR } B_1)$$

$$A < B: L = \overline{A_1} B_1 + \overline{A_0} B_1 B_0 + \overline{A_1} \overline{A_0} B_0$$

By using above obtained Boolean equation for each output, the logic diagram can be implemented by using four NOT gates, seven AND gates, two OR gates and two Ex-NOR gates.

The figure below shows the logic diagram of a 2-bit comparator using basic logic gates. It is also possible to construct this comparator by cascading of two 1-bit comparators.



[Back to top](#)

4-Bit Comparator

It can be used to compare two four-bit words. The two 4-bit numbers are $A = A_3 A_2 A_1 A_0$ and $B = B_3 B_2 B_1 B_0$ where A_3 and B_3 are the most significant bits.

It compares each of these bits in one number with bits in that of other number and produces one of the following outputs as $A = B$, $A < B$ and $A > B$. The output logic statements of this converter are

- If $A_3 = 1$ and $B_3 = 0$, then A is greater than B ($A > B$). Or
- If A_3 and B_3 are equal, and if $A_2 = 1$ and $B_2 = 0$, then $A > B$. Or
- If A_3 and B_3 are equal & A_2 and B_2 are equal, and if $A_1 = 1$, and $B_1 = 0$, then $A > B$. Or
- If A_3 and B_3 are equal, A_2 and B_2 are equal and A_1 and B_1 are equal, and if $A_0 = 1$ and $B_0 = 0$, then $A > B$.

From the above statements, the output $A > B$ logic expression can be written as

$$G = A_3 \overline{B_3} + (A_3 \text{ Ex-NOR } B_3) A_2 \overline{B_2} + (A_3 \text{ Ex-NOR } B_3) (A_2 \text{ Ex-NOR } B_2) A_1 \overline{B_1} + (A_3 \text{ Ex-NOR } B_3) (A_2 \text{ Ex-NOR } B_2) (A_1 \text{ Ex-NOR } B_1) A_0 \overline{B_0}$$

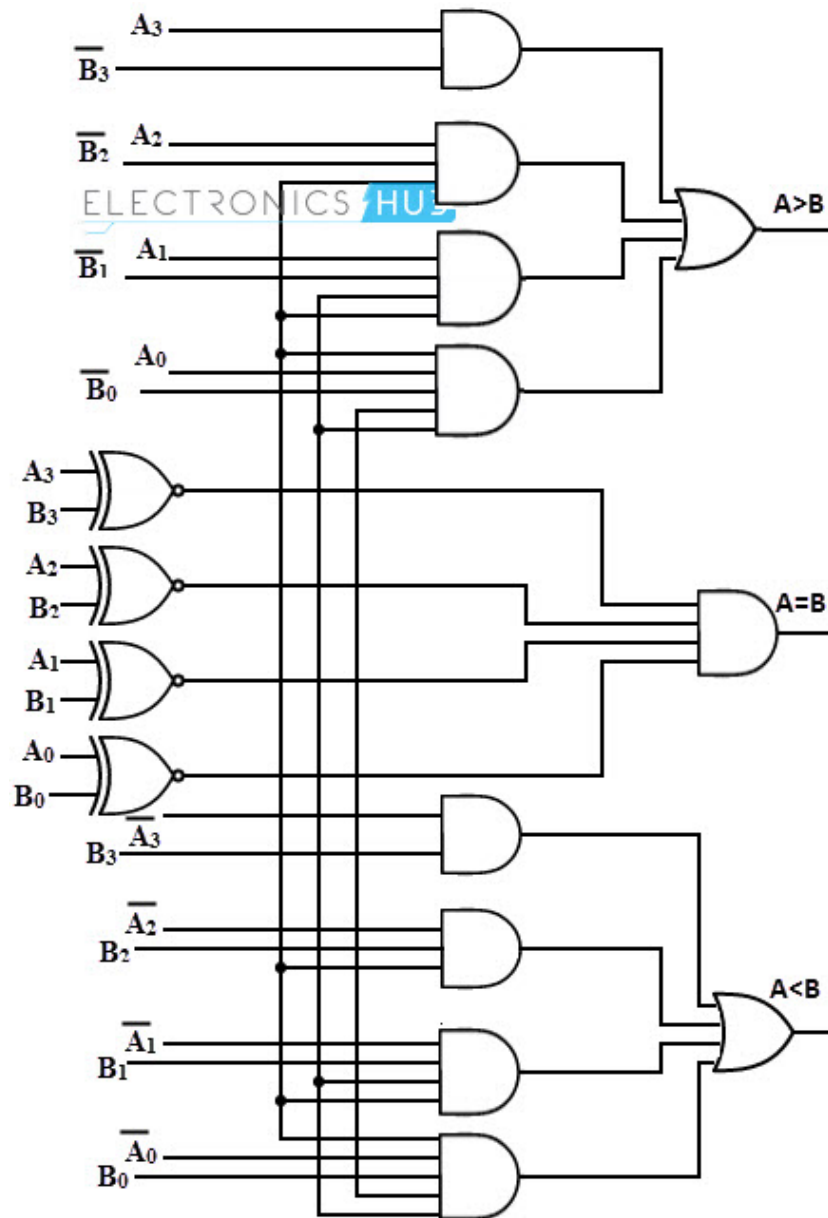
Similarly the logic expression for the L or $A < B$ output can be expressed as

$$L = \overline{A_3} B_3 + (A_3 \text{ Ex-NOR } B_3) \overline{A_2} B_2 + (A_3 \text{ Ex-NOR } B_3) (A_2 \text{ Ex-NOR } B_2) \overline{A_1} B_1 + (A_3 \text{ Ex-NOR } B_3) (A_2 \text{ Ex-NOR } B_2) (A_1 \text{ Ex-NOR } B_1) \overline{A_0} B_0$$

The equal output is produced when all the individual bits of one number are exactly coincides with corresponding bits of another number. Then the logical expression for $A=B$ output can be written as

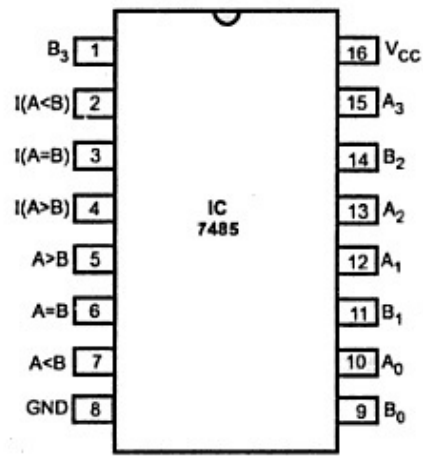
$$E = (A_3 \text{ Ex-NOR } B_3) (A_2 \text{ Ex-NOR } B_2) (A_1 \text{ Ex-NOR } B_1) (A_0 \text{ Ex-NOR } B_0)$$

From the above output Boolean expressions, the logic circuit for this comparator can be implemented by using logic gates as given below. In this the four outputs from Ex-NOR gates are applied to AND gate to give the binary variable E or $A = B$. The other two outputs are also use Ex-NOR outputs to generate the Boolean functions as shown figure.



The 4-bit comparator is mostly available in IC form and common type of this IC is 7485. This IC can be used to compare two 4-bit binary words by grounding I (A > B), I (A < B) and I (A = B) connector inputs to V_{cc} terminal. The figure below shows the pin diagram of IC7485 comparator.

In addition to the normal comparator, this IC is provided with cascading inputs in order to facilitate the cascading several comparators. Any number of bits can be compared by cascading several of these comparator ICs.



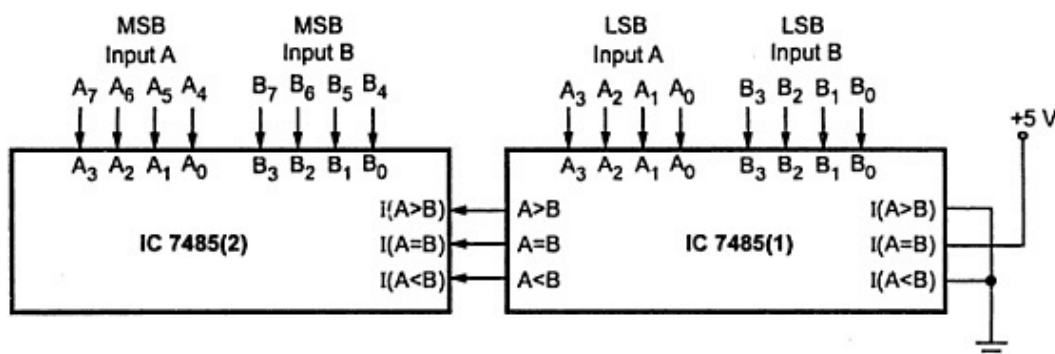
(a) Pin diagram (IC 7485)

[Back to top](#)

8-Bit Comparator

An 8-bit comparator compares the two 8-bit numbers by cascading of two 4-bit comparators. The circuit connection of this comparator is shown below in which the lower order comparator A<B, A=B and A>B outputs are connected to the respective cascade inputs of the higher order comparator.

For the lower order comparator, the A=B cascade input must be connected High, while the other two cascading inputs A, B must be connected to LOW. The outputs of the higher order comparator become the outputs of this eight-bit comparator.

[Back to top](#)

Applications of Comparators

- These are used in the address decoding circuitry in computers and microprocessor based devices to select a specific input/output device for the storage of data.
- These are used in control applications in which the binary numbers representing physical variables such as temperature, position, etc. are compared with a reference value. Then the outputs from the comparator are used to drive the actuators so as to make the physical variables closest to the set or reference value.
- Process controllers
- Servo-motor control

[Back to top](#)

FILED UNDER: [COMBINATIONAL LOGIC CIRUCITS](#)

Comments



suraj says

AUGUST 11, 2016 AT 8:54 AM

These answers are really helpful.

[Reply](#)**samra farooq says**

AUGUST 28, 2016 AT 10:07 AM

can u plz explain this relation:

$E = (L + G)'$ why u not simply use $A \text{ xnor } B$ for E in 1 bit comparator ??

[Reply](#)**Harsh Jha says**

MARCH 25, 2017 AT 6:09 AM

Very informative...

[Reply](#)**Suman says**

APRIL 22, 2017 AT 9:10 AM

Can I make a project in this magnitude comparator, by glowing LED when **AB and using 5v power supply (4-bit magnitude comparator)**.

[Reply](#)**Eel says**

MAY 19, 2017 AT 1:08 PM

Hi. thanks a lot. just one correction is needed I think: in the "Single Bit Magnitude Comparator" section the NOR gate is being used in the graphic but the article says XNOR

instead.

thanks again. really helpful.

[Reply](#)



Ikechukwu Gloria says

FEBRUARY 27, 2018 AT 5:46 PM

really helpful

[Reply](#)



Faran Ahmad says

MARCH 5, 2018 AT 2:52 PM

Thats really helpful

[Reply](#)

Leave a Reply

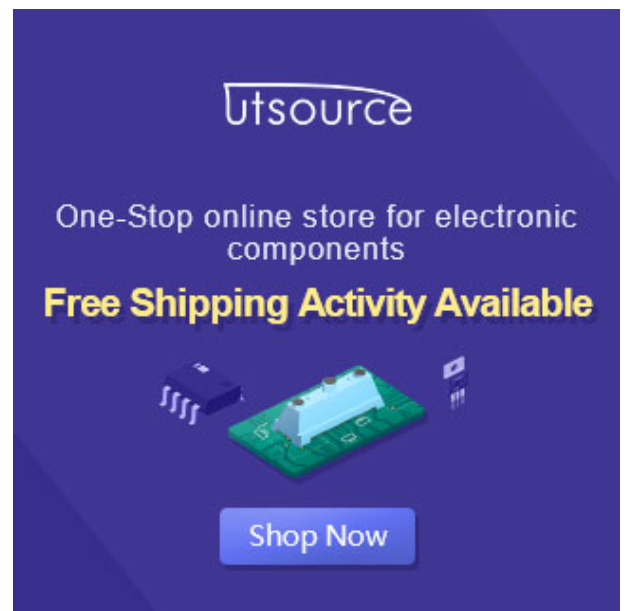
Your email address will not be published. Required fields are marked *

Comment

Name *

Email *


Website

[POST COMMENT](#)

Utsource

One-Stop online store for electronic components

Free Shipping Activity Available



[Shop Now](#)



16/18

Arduino Projects (200+)
Electronics Projects (250+)
Mini Project Circuits (160+)
Mini Project Ideas (150+)
ECE Projects (150+)
EEE Projects (150+)
8051 Projects (110+)
Raspberry Pi Projects (101+)
Electrical Project Ideas (100+)
Embedded Projects (100+)
Latest Electronics Ideas (100+)
Microcontroller Mini Projects (100+)
Robotics Projects (100+)
VLSI Projects (100+)
Solar Projects (100+)
IOT Projects (100+)

Communication Projects (70+)
LED Projects (70+)
Power Electronics Projects (60+)
RFID Projects (60+)
Home Automation Projects (50+)
Matlab Projects (50+)
EIE Projects (50+)
Wireless Projects (50+)
LabView Projects (45+)
Zigbee Projects (45+)
GSM Projects (40+)
555 Timer Circuits (40+)
Sensor Projects (40+)
ARM Projects (60+)
DTMF Projects (30+)
PIC Projects (30+)
Electrical Mini Projects (25)

KITS

[Best Drone Kits \[12\]](#)[3D Printer Kits \[12\]](#)[Best Robot Vacuum Clears \[14\]](#)**GENERAL**[Tutorials](#)[Symbols](#)[Courses](#)[Calculator](#)[Contact](#)**PROJECTS**[Electrical](#)[Electronics](#)[Embedded](#)[Power](#)[Robotics](#)[ARM](#)[IOT](#)**PROJECTS**[Mini projects](#)[Microcontroller](#)[Arduino](#)[Solar](#)[Free circuits](#)[Home Automation](#)[Seminar Topics](#)[Electronics
Questions](#)**TUTORIALS**[Capacitors](#)[Resistors](#)[Filters](#)[Diodes](#)[Transistors](#)**TUTORIALS**[Amplifiers](#)[IO Devices](#)[Thyristors](#)[DC Circuits](#)[Number System](#)**FOLLOW US**[Instagram](#)[Youtube](#)[Facebook](#)[Google Plus](#)[Twitter](#)

Copyright © 2018 [Electronicshub.org](#)