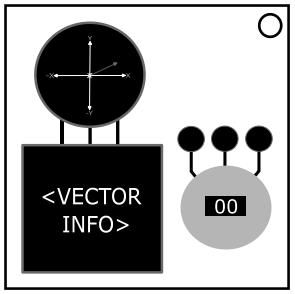
On the Subject of Vectors



Uhhh... What's with the floating 3D arrows?

This module has a 3D display of a 3-dimensional graph with the axes x, y, and z, an auto-scrolling display, and a button with a display on it.

The 3D display will have between 1-3 vectors (3D arrows) on it. The vectors will be a different color than the axes on the graph which are always white and have a rotating symbol indicating which axis they are. The auto-scrolling display will be cycling through all the data about each vector. The data values it displays are the color of the vector in the graph, the vector's magnitude, and the vector's components. The button with a display can be held down, and will show how long it has been held for on its display.

To solve this module, the button with a display must be held down for a certain amount of time and released when this time is reached. If this is done right it will count as a correct input, lighting up green LEDs and disarming the module. If the button is held for an incorrect amount of time upon release a strike will be recorded and the module WILL reset (new vectors). In order to figure out the time to hold the button down for determine how many vectors the graph has and use the corresponding section to get further instructions. If you have 1 vector, use the section '**1 Vector**'. If you have more than 1 vector use the section '**Multiple Vectors**'.

1 Vector

If 1 vector is on the graph then the auto-scrolling display will not be scrolling and will stay on the info for this vector. One of the pieces of data about this vector is missing and must be calculated (the color can never be missing). Keep in mind when performing calculations with vector components they can be positive or negative. To determine whether a component is positive or negative, reference the 3D graph. If the vector's arrow points in the direction of a negative axis, then the component is negative. Otherwise, it is positive. A vector's magnitude is always positive.

If the magnitude is missing...

Plug all known values into this equation and solve for M (rounding to the nearest tenth), where M is the missing magnitude and x/y/z are the components:
$M = \sqrt{(x^2+y^2+z^2)}$

If a component is missing...

Plug all known values into this equation and solve for the missing component (rounding to the nearest tenth), where M is the magnitude, A is the missing component, and B/C are the known components: $A = \sqrt{(M^2-B^2-C^2)}$

Now that the missing data value has been calculated another number must be received through the vector's color. Find the color in the table below and the result of the Number Calculation is the number which is needed for later along with the missing data value to perform the '**Final Calculation**'. Make sure to round this value to the nearest tenth.

Vector Color	Number Calculation
Red	(# of batteries * 5) + 3
Orange	Missing data value^3 + 16 - # of Stereo RCA ports
Yellow	((# of battery holders * 14) % 5) + 1
Green	# of RJ-45 ports + 204
Blue	8 * ((5 + vector's magnitude) + 6)
Purple	(vector's z-component + 6) % 3

Multiple Vectors

If multiple vectors are on the graph the auto-scrolling display will scroll between the different vectors data values. Vector 1 will always be the shortest and Vector 3 will always be the longest (if it exists). Use the colors of the vectors to figure out a number (rounded to the nearest tenth) that needs to be calculated for later use in the '**Final Calculation**'. Keep in mind when performing calculations with vector components they can be positive or negative, but that will never be displayed. To determine whether a component is positive or negative, reference the 3D graph. If the vector's arrow points in the direction of a negative axis, then the component is negative. Otherwise, it is positive. A vector's magnitude is always positive.

If more primary colors (RYB) appear on the vectors than secondary colors (GPO), add the magnitudes together and multiply the sum by the sum of the serial number's digits to get the necessary number.

Otherwise if more secondary colors appear on the vectors than primary colors, multiply the magnitudes together and divide that by the first digit of the serial number (divide by 1 if the first digit is 0) for the necessary number.

Otherwise if the number of primary colors on the vectors equals the number of secondary colors, square the last digit of the serial number and add the x-component of the secondary colored vector to it for the necessary number.

Make sure to add the sum of the x-components and y-components onto the necessary number and then subtract the sum of the z-components before moving on to the '**Final Calculation**'.

Final Calculation

Surrounding the hole that displays the 3D graph, a color ring will slowly fade in and out from 1 color to the next for 3 colors and then break for a while before repeating this action. This sequence of colors determines the final calculations necessary to get the button with a display's hold time.

Firstly, if only 1 vector was present, add together the number gotten from the missing data value and the number gotten from the vector's color. Otherwise, add together the necessary number calculated from '**Multiple Vectors**' and the number of Arrow modules by eXish on the bomb. This sum will be called S.

Now for each color that appears in the sequence of colors from the color ring, apply the rule it corresponds to from the bulleted list below, repeating a rule if two of the same colors appear right after one another.

- Red = Add 122 to S
- Green = Subtract the digital root of the # of modules (including needies) on the bomb from S
- Blue = Add the number of port plates and indicators on the bomb to S
- White = Subtract 27 from S

Now take the newly modified S and cut off any decimal after the whole number, take its absolute value, modulo it by 15, and add 1. S is now how many seconds the button with a display needs to be held for.

HOWEVER, if the bomb has 2 batteries, more than 2 port plates, an SND indicator, and one of the vector's colors is blue, hold the button for 0 seconds to solve the module.