THE INFURIATOR

Increase your knowledge and lose your friends!

A simple gadget this: Just a few switches, a couple of batteries and a torch globe. But it will intrigue people, annoy people — and also help you understand the basics of binary arithmetic.

What you will need to make it is six or more of the little push-on push-off switches which are commonly used on table lamps and bedlamps. As the diagram shows, these are wired in series with a couple of torch cells and an ordinary torch lamp. When all the switches are in the "on" position, the lamp will light. When any one switch is in the off position, the lamp will remain stubbornly off, no matter how much the remaining switches are operated.

To give the gadget an identity, components need to be assembled into some kind of a box, with the switches and the lamp in a line on the top and the batteries inside.

The size, style and finish of the box are left to you but it isn't a bad idea to have the name INFURIATOR on the top.

The idea is to invite your friends to manipulate the switches until the light comes on.

The annoying thing about the switches is that you can never tell, by looking at them, whether they are on or off, or whether you did or did not push them a moment ago.

This makes it particularly difficult to work through the combinations in a systematic manner; the person who fails to get the lamp on is annoyed to realise that he is pushing them in a completely random way and may just miss out repeatedly on the one correct combination.

And, if he does strike it, he knows that it was a matter of luck and he can't be sure to repeat the process.

Six switches would appear to be about the optimum number to use. Fewer than that, and the gadget isn't sufficiently infuriating. Seven or eight switches will increase its "infuriation factor" but will also make it harder for the owner to obtain the desired result in reasonable time.

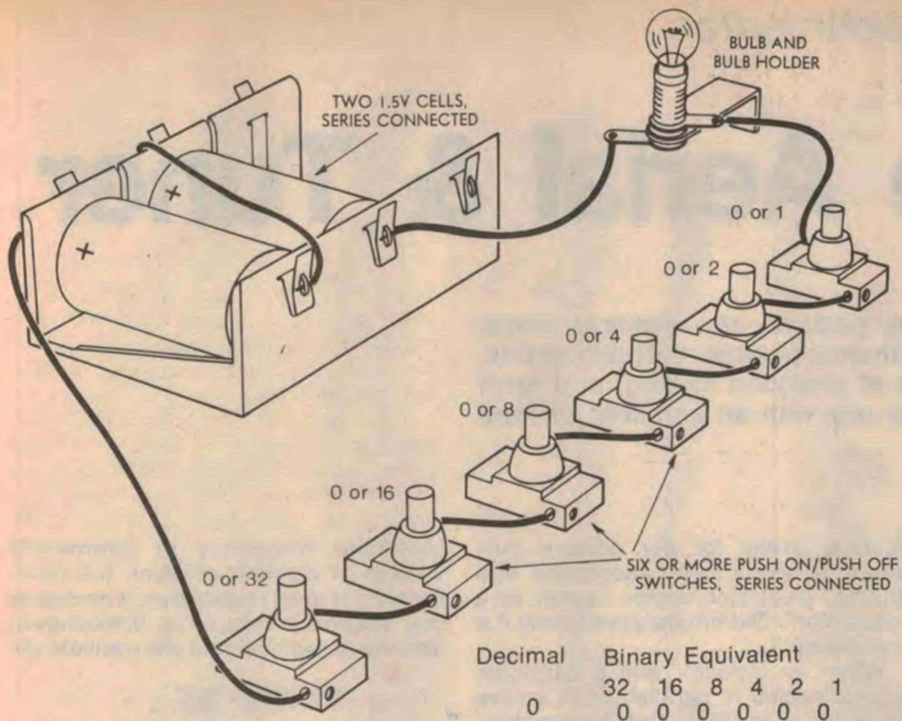
The number of possible combinations is equal to 2 raised to the same power at the number of switches. With 6 switches, the combinations add up to 2 to the power 6, which is 64. Another switch raises the figure to 128 and still another to 256.

Well then, how can one work systematically through the combinations without missing moves or duplicating them?

The answer: By treating the switches as if they were part of a binary counting system — the basis on which calculators and computers operate.

But don't tell your friends what you are up to or they will get on to the binary approach too. Then you are the one who will be infuriated!

If you know nothing about the binary system, this is a good opportunity to learn something about it. Whereas the



columns in the decimal system increase (to the left) by a factor of ten — units, tens, hundreds etc — the columns in the binary system increase by a factor of two — one, two, four, eight etc.

There are only ever two figures in a binary column; 0 or 1. The "0" is self explanatory, while the "1" represents the value assigned to that column. Thus a "1" in the four column, represents 4; a "1" in the eight column represents 8; and a "1" in both columns represents 12.

On this basis the switches can be mentally allotted column values as shown in the wiring diagram. The switch on the right has the possible values of either 0 or 1, next is either 0 or 2, the next 0 or 4, the next 0 or 8, and so on.

As the table of binary numbers show, 6 binary columns provide a code with which you can work through the 64 possible switch positions without forgetting any.

If the lamp is out and people have been randomly pushing switches, no one can know which ones are on and which ones are off. But don't worry. You simply assume that ALL switches are off and you work through the binary series from decimal 0 (binary 000000) to decimal 63 (binary 111111) and you must strike the combination that will light the lamp.

First push the right hand switch and your binary count is decimal 1.

To get decimal 2, press 1 again to bring it back to 0, then press switch 2.

To get decimal 3, press switch 1 again to re-introduce your 1 (binary 000011).

To get decimal 4, press switches 1 and 2 to bring them back to 0 and press switch 4 (binary 000100).

Decimal	Binary Equivalent					
	32	16	8	4	2	1
0	0	0	0	0	0	0
1	0	0	0	0	0	1
2	0	0	0	0	1	0
3	0	0	0	0	1	1
4	0	0	0	1	0	0
5	0	0	0	1	0	1
6	0	0	0	1	1	0
7	0	0	0	1	1	1
8	0	0	1	0	0	0
9	0	0	1	0	0	1
10	0	0	1	0	1	0

A few larger numbers in the series are shown below as examples.

15	0	0	1	1	1	1
23	0	1	0	1	1	1
41	1	0	1	0	0	1
63	1	1	1	1	1	1

To get decimal 5, press switch 1 again to add a 1 to the 4 obtained in the last step. And so on through the series.

Even if you are interrupted while doing the sequence, you can go back and continue as long as you remember the number at which you stopped. If you stopped at 23, for example, you must have had switches 16, 4, 2 and 1 pushed on. So you simply return 4, 2 and 1 to the zero state and press switch 8 to get a total of 24.

You would be extremely unlucky to have to go right through the series to 63.

It requires a certain amount of concentration to work accurately through the numbers, but it isn't all that hard.

In fact, with a little more concentration you can economise in switch operations.

If you press switches 1, then 2, then 1 again, you have covered decimals 1, 3 and 2 in that order. Now press switches 4, 1, 2, 1 in that order and you will have covered decimals 6, 7, 5 and 4. Other routines are possible and you may come up with one which you consider easier to remember.