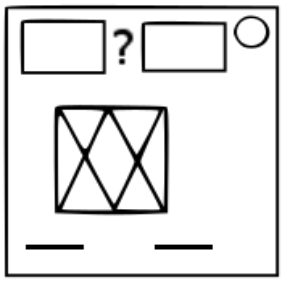
On the Subject of Boolean Wires



Flavor text

- This module contains two displays with letters on them, 8 input wires, a submit wire (green) and a reset wire (red).
- To solve the module cut the wires so the wires you didn't cut form the symbol of a logical operator which makes the statement true (Letter reference on page 2).
- Valid operators are: OR; XOR; AND; NAND; NOT (Operators on page 3).
- You can only use the OR and the NAND operator once per module!
- After 5 correct answers the module will disarm.
- Entering an incorrect or invalid symbol will result in a strike and the module'll reset (And you can use the OR and NAND operators again).

Letter reference

- An "!" on the module means NOT (True becomes false and vice versa.)

Letter	True if...		Letter	True if...
A	there's a vowel in the serial #.		N	there's a duplicate port on the bomb.
B	more than 1 D battery.		O	unlit MSA indicator present.
C	last digit of the serial # is even.		P	lit FRQ indicator present.
D	even number of modules on the bomb.		Q	true
E	there's a Forget Me Not on the bomb.		R	number of modules is divisible by 3.
F	lit BOB indicator present.		S	more modules than starting time in minutes.
G	unlit CAR indicator present.		T	more than 71 modules.
H	no batteries present.		U	solved modules > unsolved modules
I	exactly 1 pair of AA batteries on the bomb.		V	number of solved modules is even.
J	number of modules on the bomb is prime.		W	PS/2 port present.
K	less than half of the bomb's original time left.		X	RCA port present.
L	Parallel port present.		Y	number of lit indicators = number of unlit indicators
M	Serial port present.		Z	no indicators present.

Operator reference

- There are symbols you can enter. If there are more operators that make the statement true, either can be used.
- bt1 refers to byte 1; bt2 refers to byte 2. (A.K.A. First and second letter)
- In the table if a byte is in "**not()**" that means that byte is false. Otherwise the byte is true.
- **Reminder:** You can only submit OR and NAND once!

Symbol and name	Makes statement true if...
∨ (OR)	• bt1 ∨ not(bt2) • not(bt1) ∨ bt2 • bt1 ∨ bt2
⊻ (XOR)	• bt1 ∨ not(bt2) • not(bt1) ∨ bt2
∧ (AND)	bt1 ∧ bt2
| (NAND)	• bt1 ∨ not(bt2) • not(bt1) ∨ bt2 • not(bt1) | not(bt2)
¬ (NOT)	not(bt1) ¬ not(bt2)

- **Note:** The NOT operator in this module (¬) is not an actual NOT operator and it isn't used for telling if both input is false.