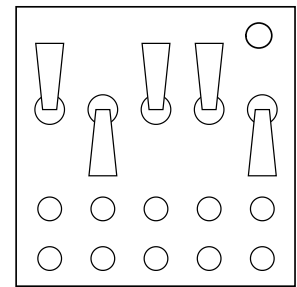# On the Subject of Uncolored Switches

Oh no, it's one of these module chains again.

Oh no, it's one of these module chains again.

- The module has five black switches and 10 LEDs beneath them. If the switches are not black or the LEDs do not have color, you are looking at a different module.
- Each LED can be orange, green, purple, turqoise, blue, red, white, or unlit.
- To disarm the module, set the switches to their preferred positions each stage for three stages.
- For each switch, use the colors of the two LEDs directly below it with Table A to obtain two True/False inputs.
- Observe the current positions of the switches and locate its assigned logic operator in Table B.
- Apply the operator to each of the five sets of inputs. If the result of the operation for a switch is True, that switch's preferred position is up. If the result is False, the preferred position is down.
- In order to submit a pattern, press the switches to change their position from up to down. 2 seconds after a switch is flipped, the solution will submit unless another switch is flipped.
- If an invalid solution is submitted, a strike will be incurred and the switches will reset to their initial positions.

Table A: LED Color Reference:

Orange	Green	Purple	Turquoise
There are more batteries than there are ports + indicators.	The switches form an illegal position from the Switches manual.	There is a Souvenir, Turn the Key, Needy module, or any module with "Forget" in its name.	The number of red and blue LEDs is greater than the number of orange and purple LEDs.
Red	**Blue**	**White**	**Unlit**
There is at least one strike, or there is a Two Factor.	There is a lit NSA or FRK indicator, or there is an unlit MSA or FRQ	There are at least 3 switches that point upwards.	There is at most 1 white LED.

Table B: Position to Operator:

Logic Operator	Representative switch states				Operator explanation
AND					The result is true if both of the inputs are true.
OR					The result is true if at least one of the inputs is true.
XOR					The result is true if only one of the inputs is true.
NAND					The result is true if at least one of the inputs is false.
NOR					The result is true if both inputs are false.
XNOR					The result is true if both inputs share the same state.
Implication Left / Implies					The result is false if top input is true and bottom input is false. Else returns true.
Implication Right / Is implied by					The result is false if top input is false and bottom input is true. Else returns true.