Oxyll elegant Bluetooth mod (part 3)

February 2021

Below is more information on the Bluetooth mod that I put together for a 5th gen iPod Classic to enable Bluetooth playback on the iPod, without the need for modifications to the outer shell of the iPod - no holes, missing screws, or cutting required. More than anything, this is a tutorial on how logic gates work. If you have questions, you can more than likely find the answer with some searching on logic gates.
Soldering is required although that shouldn't be surprising to any of you reading this.

Original showcase of the mod:
https://www.reddit.com/r/IpodClassic/comments/iymhrq/i_previously_eluded_to_an_elegant_way_to_do_the/

My first attempt to explain the mod:
https://www.reddit.com/r/ipod/comments/j81zo8/my_elegant_bluetooth_mod_part_2_details/

Hi, it's oxyll, https://www.reddit.com/user/oxyll/

How (simple) **buttons work**

Components:

- Button
- Sense IC (controller, transistor, etc)
- Power source, ground
- Resistor

For this example,

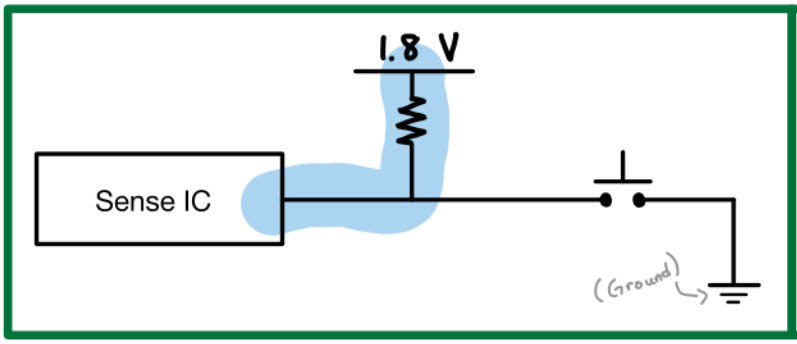
- Power (Vcc) will be 1.8V.
- Ground is 0V.
- “Digital 1” will be equal to Vcc.

2 options for simple buttons (with 2 possible states for each design)

Active low (pull-up resistor)

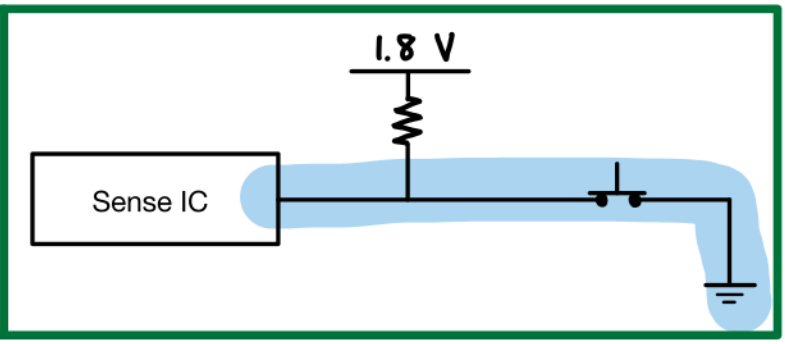
Hi, it’s oxyll

Button not pushed



The IC pin will measure 1.8V because it is pulled up by Vcc through the resistor.

Active low (pull-up resistor). Button pushed



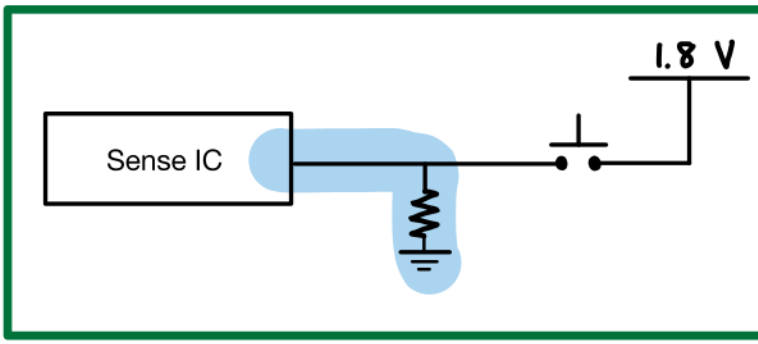
The IC pin will measure 0V because it is directly connected to GND through the switch.

Hi, it’s oxyll

Active high (pull-down resistor)

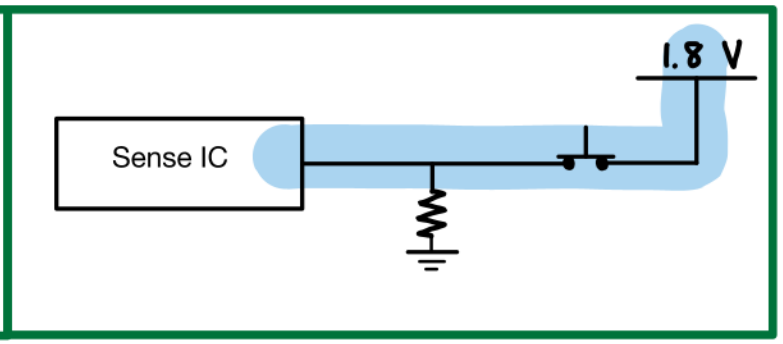
Hi, it’s oxyll

Button not pushed



The IC pin will measure 0V because it is connected to GND through the resistor.

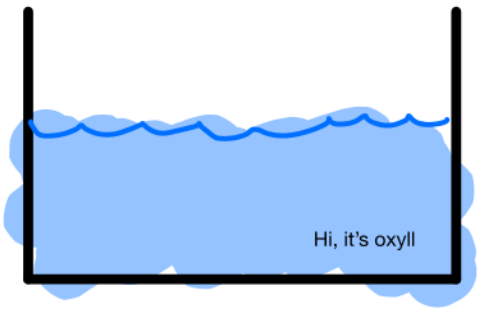
Button pushed



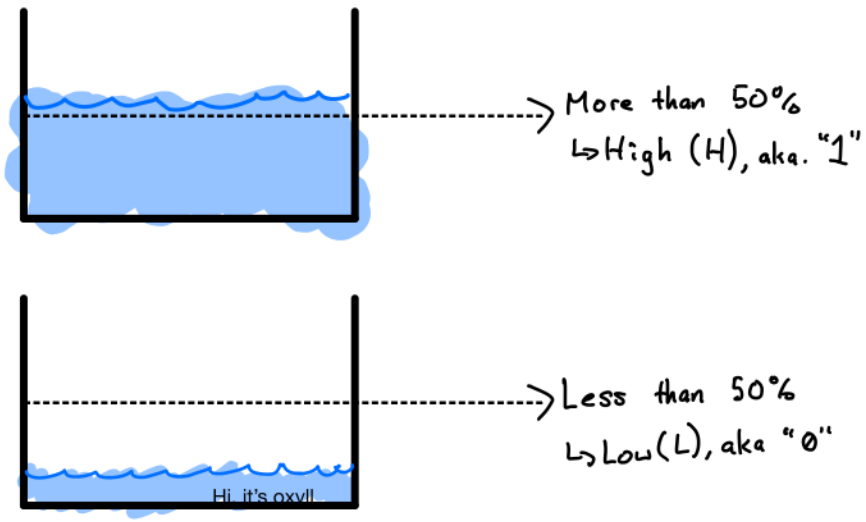
The IC pin will measure 1.8V because it is connected to Vcc through the switch.

A practical analogy on how active high / active low buttons work

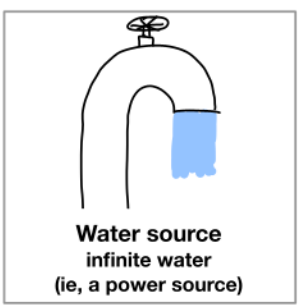
The premise: Here is a water tank.

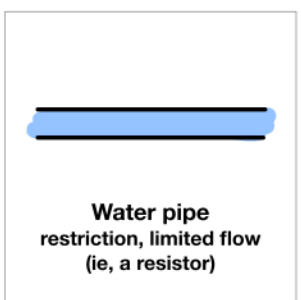


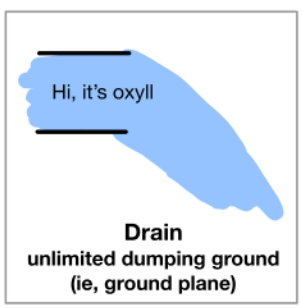
The water tank has a water level sensor on it. All this sensor cares about, is whether the water level is above 50%, or less than 50%. (If you really want to know how this would work, search up how fuel tank sensor work in cars)

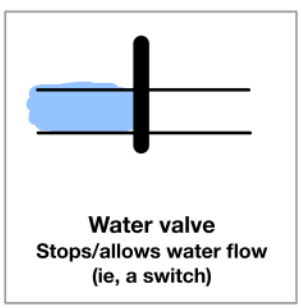


There is stuff attached to the tank to make the button circuitry work





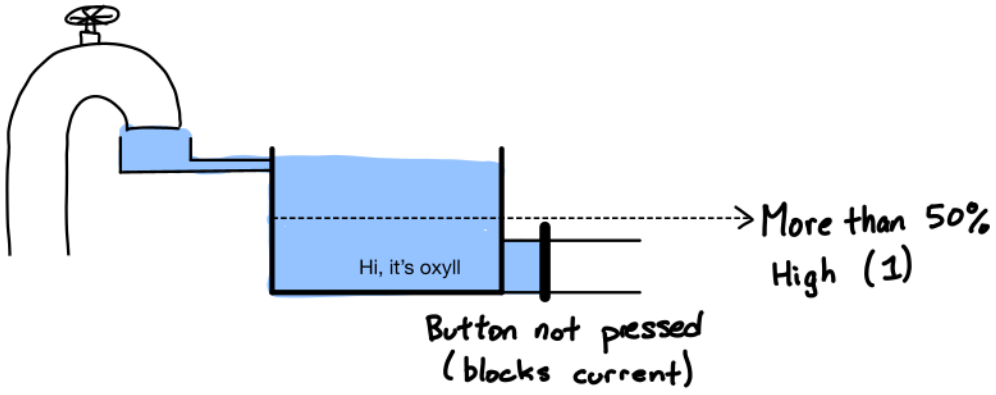




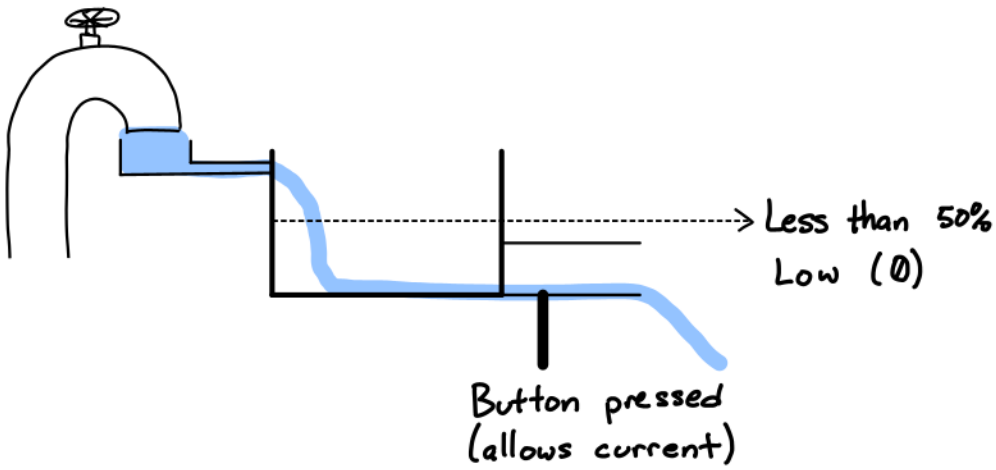
Yes, negative voltage exists, but we won't worry about that here.

Hi, it's oxyll

Active low (pull up resistor)

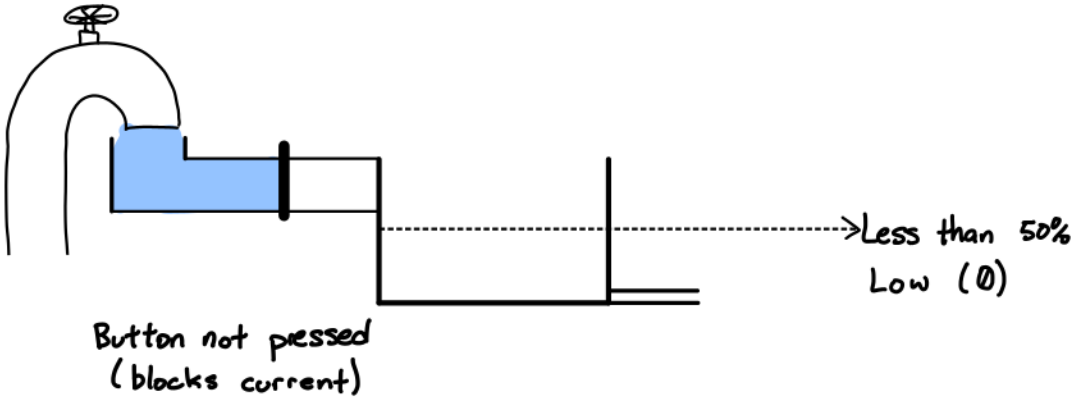


When the button is not pressed, the small pipe (resistor) allows the water to slowly fill up the tank until it is full.
The level sensor reads high.
The water level is *pulled up* by the resistor. —> pull up resistor

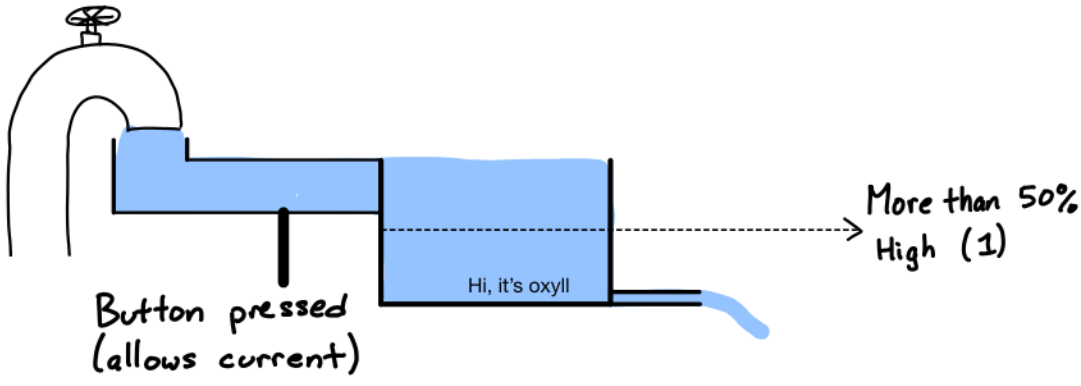


When the button is pressed, the large pipe quickly empties the tank, and the water from the small pipe still flows but is instantly emptied by the large pipe.
The level sensor reads low.
The water level is *low* when the button is *activated*. —> active low

Active high (pull down resistor)



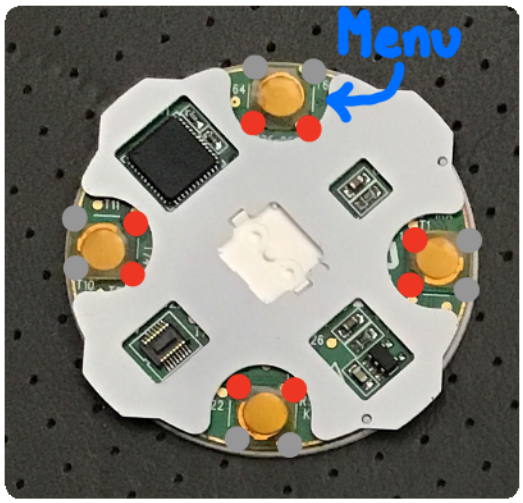
When the button is not pressed, water does not enter the tank, and any water in the tank is drained away by the small pipe.
The level sensor reads low.
The water level is *pulled down* by the resistor —> pull down resistor

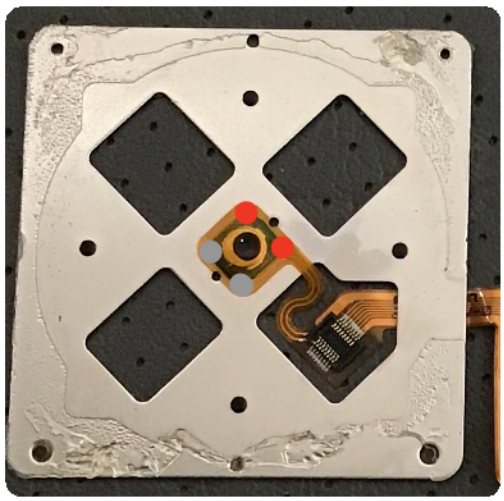


When the button is pressed, water flows into the tank, and quickly fills it. The small pipe is constantly draining water but the tank still filled because of the much larger inlet pipe.
The sensor reads high.
The water level is *high* when the button is *activated* —> *active high*

4th gen Monochrome - active low face buttons

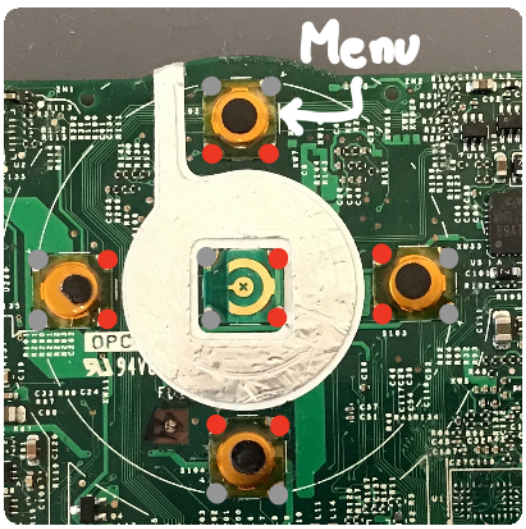
(iPod Photo is probably is the same..?)

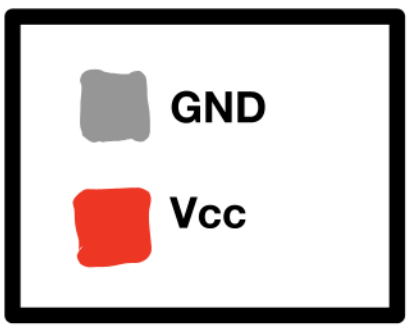




Hi, it's oxyll

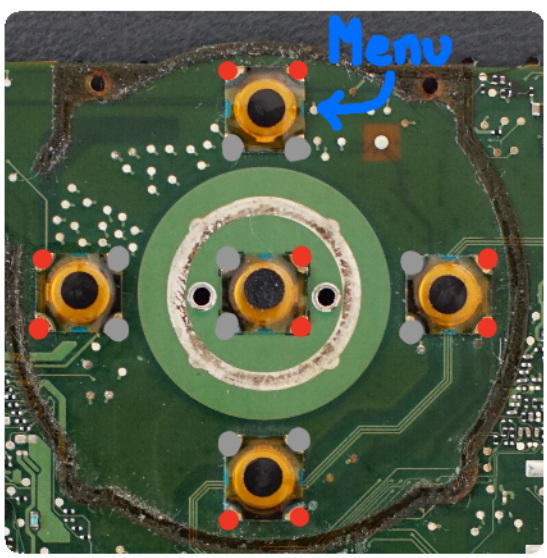
5th gen - active low face buttons





Hi, it's oxyll

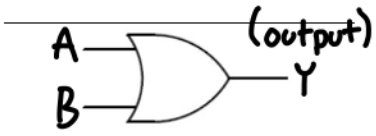
6th gen - active low face buttons



Gates

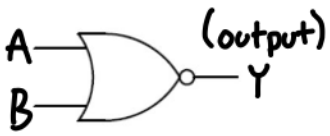
You should only need EITHER an OR gate or a NOR gate for the Bluetooth mod. The one you need depends on whether the button on your bluetooth transmitter uses an active low (OR gate), or an active high (NOR gate).

If you want to be pedantic, you should be watching for the threshold of when a voltage becomes digital 1 and when it becomes digital 0. This information will be on the datasheet for the logic gate, and you will need to measure the voltage on the iPod and on your transmitter to ensure that they are fall within acceptable values as stated on the datasheet. If you're lazy... just buy any one of the correct gate type (OR vs NOR) and it will probably work.



OR

A	B	Output
0	0	0
0	1	1
1	0	1
1	1	1



NOR

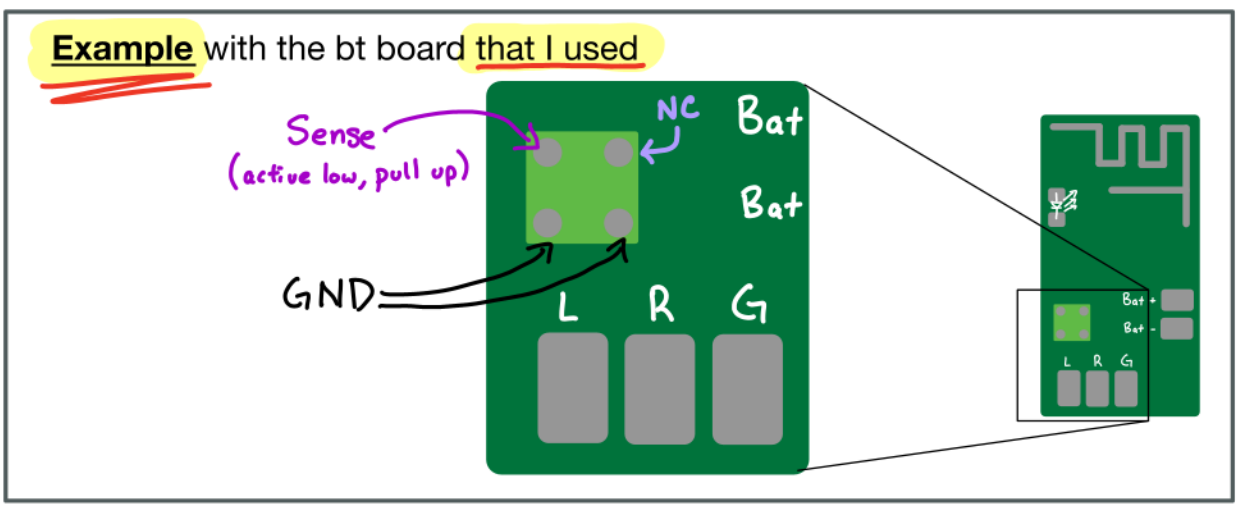
A	B	Output
0	0	1
0	1	0
1	0	0
1	1	0

The iPods has active low face buttons, so when they are not pressed, the corresponding sense pin will see a digital 1. When the button is pressed, the sense pin will be digital 0.

If your transmitter uses an **active low schema, use an OR gate.**
It expects a digital 0 when the transmitter button is pressed; to substitute the button, an OR gate has the behaviour needed (input of two 0 results in 0).

Hi, it's oxyll

If your transmitter uses an **active high schema, use a NOR gate.**
It expects a digital 1 when the transmitter button is pressed; to substitute the button, a NOR gate has the behaviour needed (input of two 0 results in 1).



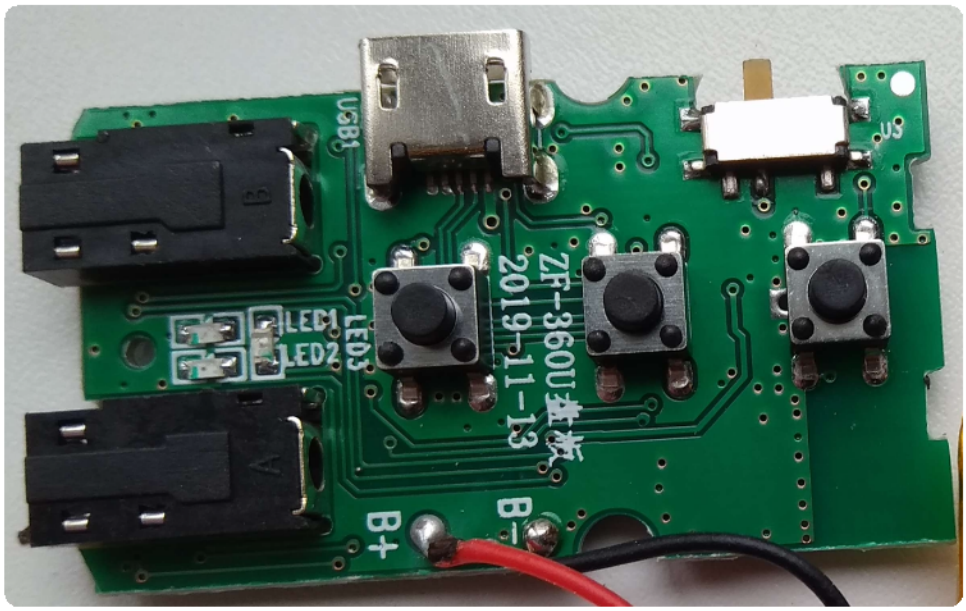
How do you know if your transmitter uses an active high or active low schema?

Use a multimeter to measure the voltage of the pins while the button is not pressed, and when it's pressed.

Hi, it's oxyll

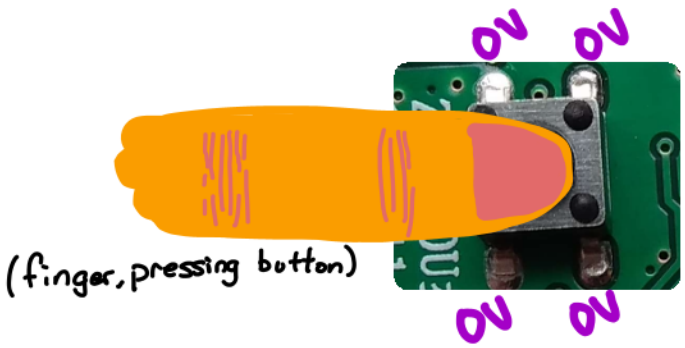
So practically, how do you determine whether the the logic is active high or active low? Example with the one BT board I used.

The middle button is the power button and the pairing button, so that is the only button that matters.



If you measured the voltage when the button is NOT pressed, you may get something similar the following

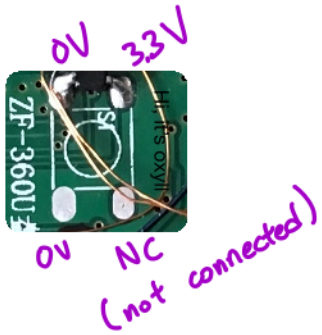
With the button pressed, you may get the following



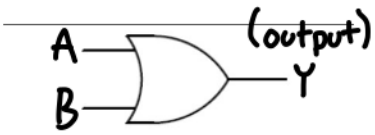
The side that didn't change (left, in this case) doesn't matter much. The other side (right, in the case), is high when not pressed, and low when pressed —> active low.
The right side changes voltage, so the sense circuitry must be connected to there.

Once you remove the button (though you can choose to leave it on), you may find that only one of the legs of the button connected to the sense circuitry will act weird. This is because it is not connected to anything, and is a floating pin. Just ignore it and use the other pin.

In this case, that means the top right pin is the one we need to connect the output of the logic gate.



We now know which pin we need to connect the output of the gate to. Now, which gate to use? The ipod face buttons are active low, which means they provide a 0 when pressed. This BT board needs a 0 to emulate the button being pressed. If either ipod face button is not pressed, then the output of the gate needs to be 1 (to emulate the button not being pressed).



OR

A	B	Output
0	0	0
0	1	1
1	0	1
1	1	1

The OR gate matches our needs. The output is only 0 when both inputs are 0, meaning both ipod face buttons are pressed.

Putting it all together

This example uses the Select and Next buttons to simulate a button press on the transmitter. If you want the button combination to be more than 2 buttons, you can either use a gate with more inputs (more compact, easier to solder), or use multiple gates (probably cheaper).

If you want to use a different button combination, simply choose the correct pins on the board to connect to inputs A and B on the gate. For this project, it does not matter which button connects to A and which to B.

If you have questions, try to find the answer by searching for it, and if you're still stuck after that, you can find me on reddit: u/oxyll