The Burglar Alarm

When I was a kid, my grandmother owned a liquor store. I was fascinated by the burglar alarm. We set it when we locked up, and turned it off when we opened the store. It was monitored by the Holmes agency, which would respond if there was a problem.

It had several sensor circuits, which worked in different ways.

1. In the basement where there was a great deal of storage, there was a thin black string near the floor. It was pulled taut. If someone knocked the string with their foot, it snapped a switch which triggered the alarm. The same switch triggered the alarm if the string was broken.
2. There was an aluminum tape on the windows. If the glass was broken, it would break the aluminum trace which would also trigger the alarm. Some windows, which could be opened, had contacts which would be broken if the window were opened.

This system actually requires two different kinds of electrical circuitry.

1. One circuit is normally open. Closing the circuit triggers the alarm.
2. The other circuit is normally closed, opening it triggers the alarm.

Having two different types of circuit can confuse the burglar who, seeing a switch (perhaps a switch on a door or window) or even the wires going to the switch is not sure how to bypass it to avoid detection.

Remember this was many years ago, alarm systems can be FAR more sophisticated now.

So here is the idea for the burglar alarm circuit:

One entire set of sensors is simulated by a single normally open push button. Pushing the button closes the circuit. There could be multiple sensors wired in parallel. Activating any one of them will activate the circuit.

Start with that.

Find the schematic diagram for a pushbutton (you used TWO of them in the fire alarm).

1. Draw a neat, proper schematic with one pushbutton.
2. Construct the Blockly Prop code which will react as follows

a) When you start, the terminal displays “System Normal”
b) A green LED is on, flashing on/off
c) If the button is pushed, the terminal displays “PROBLEM”
d) If the button is pushed, the green light goes off, the red light flashes, and the Piezo speaker beeps intermittently (beep, pause, beep, pause…)

Download to the EEPROM. That way, the blue reset button will cause the program to start from the top and you do not need to program a “clear alarm” or “restart” routine.

3. Add to the previous schematic another identical circuit, but replace the button with a specific color wire.   Let’s say a YELLOW wire, and do not use a yellow wire anywhere else in the circuit, just to make troubleshooting easier.
4. Add this circuit to your breadboard.
5. Construct the blocklyprop code which will react as follows:

a) When you start, the terminal displays “System Normal”
b) A green LED is on, flashing on/off
c) If the yellow wire is broken, the terminal displays “PROBLEM”
d) If the yellow wire is broken, the green light goes off, the red light flashes, and the Piezo speaker beeps intermittently (beep, pause, beep, pause…)

Download to the EEPROM. That way, the blue reset button will cause the program to start from the top and you do not need to program a “clear alarm” or “restart” routine.

6. Now modify your program to combine both functions and be prepared to demonstrate it.
7. There are any number of enhancements possible. For example, when you turn it on there could be a delay of some number of seconds before it was actually active, to give you a chance to leave the store if the front door was alarmed.
8. There could be a delay when you entered by the door during which you had some seconds to enter a secret code before the alarm was triggered.
9. The terminal could be more specific about what was triggered.