Defuse the Bomb
A CSC 102 Project

Team: Justin Gaalswijk, Joey Parsons, Joshua Tucker

BOMB DEFUSAL MANUAL

The Game

This project is based on the game **Keep Talking and Nobody Explodes**, a cooperative bomb defusing party game. As the designers put it, “You’re alone in a room with a bomb. Your friends, the ‘Experts’, have the manual needed to defuse it. But there’s a catch: the Experts can’t see the bomb, so everyone will need to talk it out - fast! Put your puzzle-solving and communication skills to the test as you and your friends race to defuse bombs quickly before time runs out!”

Their version is a software game. Our version takes the idea and realizes it as a physical hardware device with buttons, switches, and more! Although our version can be played just like theirs, players can interact with both the bomb and this document at the same time (i.e., players can simultaneously defuse the bomb and serve as the “Experts”, using this document to help disarm the phases). The backend of our version of the game is a Raspberry Pi computer that combines a typical computer with the ability to interact with the outside world through sensors. The underlying software is written in Python and is the result of a final group-based project in CSC 102 (The Science of Computing II) in the Computer Science Program at the University of Tampa.

Defusing Bombs

The bomb will ‘explode’ when its countdown reaches 00:00 or when too many strikes have occurred. You defuse the bomb by disarming all of its phases before the countdown expires.

Phases

The bomb has four phases, each of which must be disarmed to defuse the bomb. The phases can be disarmed in any order. Once a phase is disarmed, it becomes inactive, and changing it doesn’t affect the bomb. Instructions for disarming the phases are provided in this document.

Strikes

A mistake in disarming a phase results in a strike. Get too many strikes, and the bomb”” explodes”. Sometimes, the remaining countdown time will be decreased and/or tick faster when a certain number of strikes have occurred.

Information

A random version of the bomb is presented each time it is “booted”, with a whopping 45,158,400 possible variations!
Disarming some phases will require specific information about the bomb. Pay close attention to the “bootup” text on the bomb’s screen

---

Regarding the Toggles

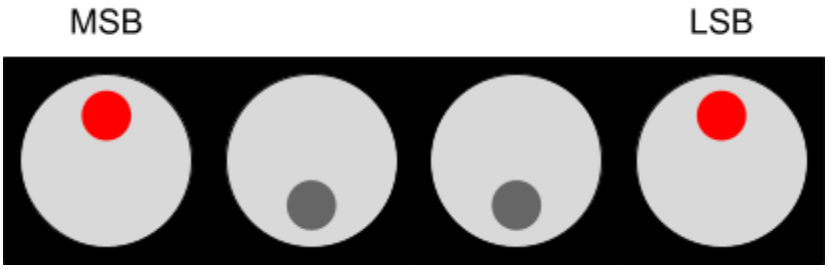
The correct state of each toggle switch is based on the bomb's serial number.

You must first add the numeric digits in the serial number together to obtain a target value. Convert this value to a 4-digit binary number. Toggle the switches to represent the binary number.

Converting a decimal number (base 10) to a binary number (base 2) can be done by placing a 1 in the appropriate powers of two represented by the columns of the table below that, when added together, sum to the value. A 0 is placed in the remaining columns. The leftmost digit of the binary number is known as the most significant bit, while the rightmost digit is known as the least significant bit

2^3	2^2	2^1	2^0
8	4	2	1
x	x	x	x

The LED on a toggle switch lights up to represent a binary 1. For example, the diagram below represents 1001 in binary, or 9 in decimal.



Regarding the Button

The button can be pressed at any time while defusing the bomb, but releasing your hold on the button has to be done at a specific time.

The button will be colored red, blue, or green, which means something different for when you have to release it.

Red	Release at any time
Blue	Release when the last digit in the serial number is anywhere in the seconds left on the timer
Green	Release when the last digit in the serial number is anywhere in the seconds left on the timer

Regarding the Keypad

The keypad phase challenges you to convert a **hexadecimal number** (base-16) into a **decimal number** (base-10).

The LCD will display a message like:

`1F →`

Your goal is to **press the corresponding decimal digits** using the keypad.

- For `1F`, the correct decimal is `31`
- You must press `3`, then `1` on the keypad

If the full input matches the correct decimal value:

- The phase will be **defused**
- If incorrect, you will receive a **strike**

Note: You can only enter a number once per attempt. If it's wrong, the game resets that phase for another try.

Regarding the Wires

The correct wire configuration is based on a **5-bit Two's Complement binary number**, which is shown on the bomb's screen as the *Target*.

You must match the bit pattern shown by **connecting or disconnecting the 5 jumper wires** accordingly.

Each jumper wire corresponds to one bit in a 5-digit binary number. The bits are read from **left to right**, with:

- The **leftmost wire** is the **most significant bit (MSB)**
- The **rightmost wire** is the **least significant bit (LSB)**

A wire that is **plugged in** represents a **1**, while a wire that is **left unplugged** represents a **0**.

> Example: If the target is `-3`, the screen will show:
>
> ```
> Wires: 11101 (= 29) | Target: -3
> ```

You must configure the wires like this:

Wire Position	Bit Value	Connection
Wire 1 (MSB)	1	Plugged in
Wire 2	1	Plugged in
Wire 3	1	Plugged in
Wire 4	0	Unplugged
Wire 5 (LSB)	1	Plugged in

The phase will automatically defuse once your physical wire setup matches the bit pattern.

Do note that the wires will show your regular binary equivalent, not the 2's complement.
