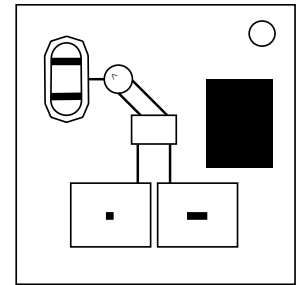


## On the Subject of Encrypted Morse

*Oh nice, a Morse module! Wait, what does HNWLI REAKHI mean?*

- This module contains a **Morse light** that transmits one of two things: The encrypted **message** (orange light) or the **key** (blue light). The defuser can choose which of these to receive using the knob next to the Morse light.
- There is also a **binary display** that shows a sequence of 42 binary digits (**sequence A**).
- The defuser may at any time press the small reset button in the middle of the module to restart the transmission from the beginning, as well as erase all inputted symbols.
- The wires leading to the Morse light may also be pressed to disable/enable it.



### Step 1: Generating sequence B

For every character of the serial number, generate a 7-digit binary number using the following rules, then concatenate them into a 42-digit binary string. That is **sequence B**.

- If the character is a digit, use its binary representation, extended to 7 digits. Example: 2 = 0000010 and 5 = 0000101.
- If the character is a vowel, use 1011001.
- Otherwise, if the character is positioned in the alphabet before the letter O, use 1100010.
- Otherwise, use 1000100.

If the total number of batteries on the bomb equals the total number of ports, reverse the first half of sequence B.

### Step 2: Binary decryption

Treat the received Morse message as a continuous sequence of Morse symbols (• or -), omitting any spaces between letters.

For every symbol in that sequence, let A be the binary digit from sequence A at the same position. B is the corresponding digit from sequence B.

Use that information with the following table to decrypt the symbol. "Space" refers to a space between Morse letters.

Binary digits		Received symbol	
A	B	.	-
0	0	Space	-
0	1	-	.
1	0	.	-
1	1	.	Space

Parse the decrypted Morse message to get your intermediate result.

If this intermediate message contains more than one vowel, swap the first and fifth characters and the second and eighth characters in both the intermediate message and the received key.

### **Step 3: Alphabetic decryption**

For every letter in your intermediate result:

- Take the letter from the received key at the same position.
- Add the alphabetic positions of your message letter to the alphabetic position of the key letter (A=1, Z=26).
- If the result is greater than 26, subtract 26.
- Convert the result to the letter at that alphabetic position.

You now have the fully decrypted message.

### **Step 4: Response**

Find the correct response to send in the following table. Note that spaces and punctuation are added for readability, they are irrelevant in both message and response.

Message	Response	Message	Response
"DETONATE"	"PLEASE NO"	"REMEMBER"	"SOUVENIR"
"READY NOW"	"CHEESECAKE"	"GREAT JOB"	"THANK YOU"
"WE'RE DEAD"	"SADFACE"	"SOLO THIS"	"I DARE YOU"
"SHE SELLS"	"SEA SHELLS"	"KEEP TALK"	"NO EXPLODE"

To defuse the module, convert the correct response to Morse code and use the two large buttons on the module to transmit it. Spaces between words and letters are not required. Transmitting a wrong Morse symbol at any time will cause a strike and you will have to restart the transmission from the beginning.

**Appendix A: Morse Code and binary conversion****How to Interpret**

1. A short flash represents a dot.
2. A long flash represents a dash.
3. There is a long gap between letters.
4. There is a very long gap before the word repeats.

A ● ■  
 B ■ ■ ● ● ●  
 C ■ ■ ■ ●  
 D ■ ■ ● ●  
 E ●  
 F ● ● ■ ■ ●  
 G ■ ■ ■ ●  
 H ● ● ● ●  
 I ● ●  
 J ● ■ ■ ■ ■  
 K ■ ■ ● ■ ■  
 L ● ■ ■ ● ●  
 M ■ ■ ■  
 N ■ ■ ●  
 O ■ ■ ■ ■  
 P ● ■ ■ ■ ■ ●  
 Q ■ ■ ■ ● ■ ■  
 R ● ■ ■ ●  
 S ● ● ●  
 T ■ ■

U ● ● ■ ■  
 V ● ● ● ■ ■  
 W ● ■ ■ ■ ■  
 X ■ ■ ● ● ■ ■  
 Y ■ ■ ● ■ ■ ■ ■  
 Z ■ ■ ■ ■ ● ●

1 ● ■ ■ ■ ■ ■ ■  
 2 ● ● ■ ■ ■ ■ ■  
 3 ● ● ● ■ ■ ■ ■  
 4 ● ● ● ● ■ ■ ■  
 5 ● ● ● ● ●  
 6 ■ ■ ● ● ● ●  
 7 ■ ■ ■ ● ● ●  
 8 ■ ■ ■ ■ ● ●  
 9 ■ ■ ■ ■ ■ ●  
 0 ■ ■ ■ ■ ■ ■

Digit	Binary
0	0000000
1	0000001
2	0000010
3	0000011
4	0000100
5	0000101
6	0000110
7	0000111
8	0001000
9	0001001