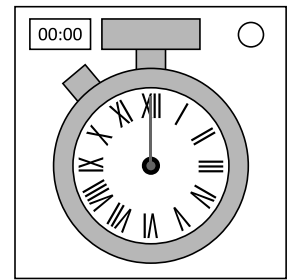


## On the Subject of The Stopwatch

*Patience is key!*



- The module consists of a stopwatch with a start/stop button on it (located at the top left), and a display which shows the time it was started/stopped. The timer display will go green when the stopwatch is activated.
- To defuse the module, you must calculate and run the stopwatch for the required amount of time. Stopping after an incorrect amount of time will cause a strike and reset the stopwatch to 0.
- **Warning: if the time remaining on the bomb runs to less than the amount of time the stopwatch must run for, the module will become unsolvable.**
- If the start time of the bomb is between 1:01 and 5:00 minutes, the required stopwatch runtime in seconds should be divided by 10 and rounded down if it is initially greater than 30 seconds.
- If the start time of the bomb is 1:00 minute or less, the required stopwatch runtime in seconds should be divided by 20 and rounded down if it is initially greater than 30 seconds.
- The timer display should be used as a guide only. The stopwatch will always count true seconds and must be run for the required amount of true seconds.

### If the serial number has 2 digits:

Create the two possible two digit numbers from the serial number.  
Multiply the digits and subtract it from each of the two numbers.  
Order the numbers from smallest to largest.  
Take the first number modulo 3. Take the second number modulo 4.  
Compare the results with *Table 1* to find the correct runtime.

### If the serial number has 3 digits:

Digits 1, 2 & 3 will be referred to as X, Y & Z respectively.  
If  $XY > ZY - X$ , increase X, Y & Z by the number of batteries on the bomb.  
If exactly two digits are even, ignore the odd value.  
Otherwise, if exactly two digits are odd, ignore the lowest value.  
Otherwise, if all three are even, ignore value X.  
Otherwise, increase value Y by two and ignore the lowest value.  
Compare the remaining numbers with *Table 2* to find the correct runtime.

### If the serial number has 4 digits:

If the second or fourth digit is 0, increase it by 1.  
Divide the first digit by the second. Divide the third digit by the fourth.  
If an integer value is not yielded, add the digits instead.  
Take both numbers modulo 4.  
Compare the results with *Table 1* to find the correct runtime.

**Table 1**

	First Number				
Second Number		0	1	2	3
	0	4:20	1:06	2:44	2:32
	1	1:13	3:14	1:39	3:22
	2	1:56	2:38	4:00	3:15
	3	4:29	3:24	2:01	0:01

**Table 2**

	First Number		
Second Number		Odd	Even
	Odd	1:27	4:12
	Even	2:35	3:40