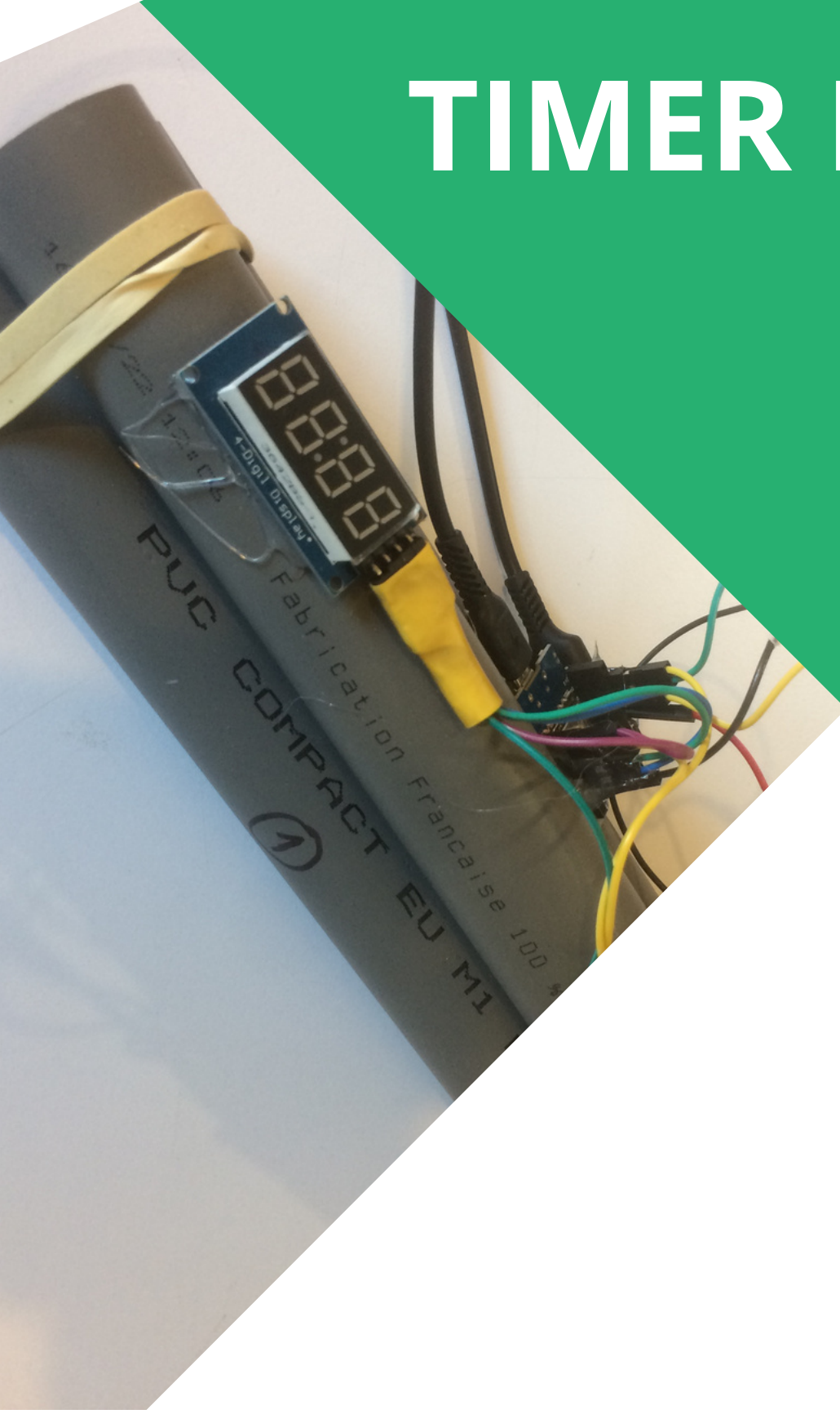




TIMER BOMB



01

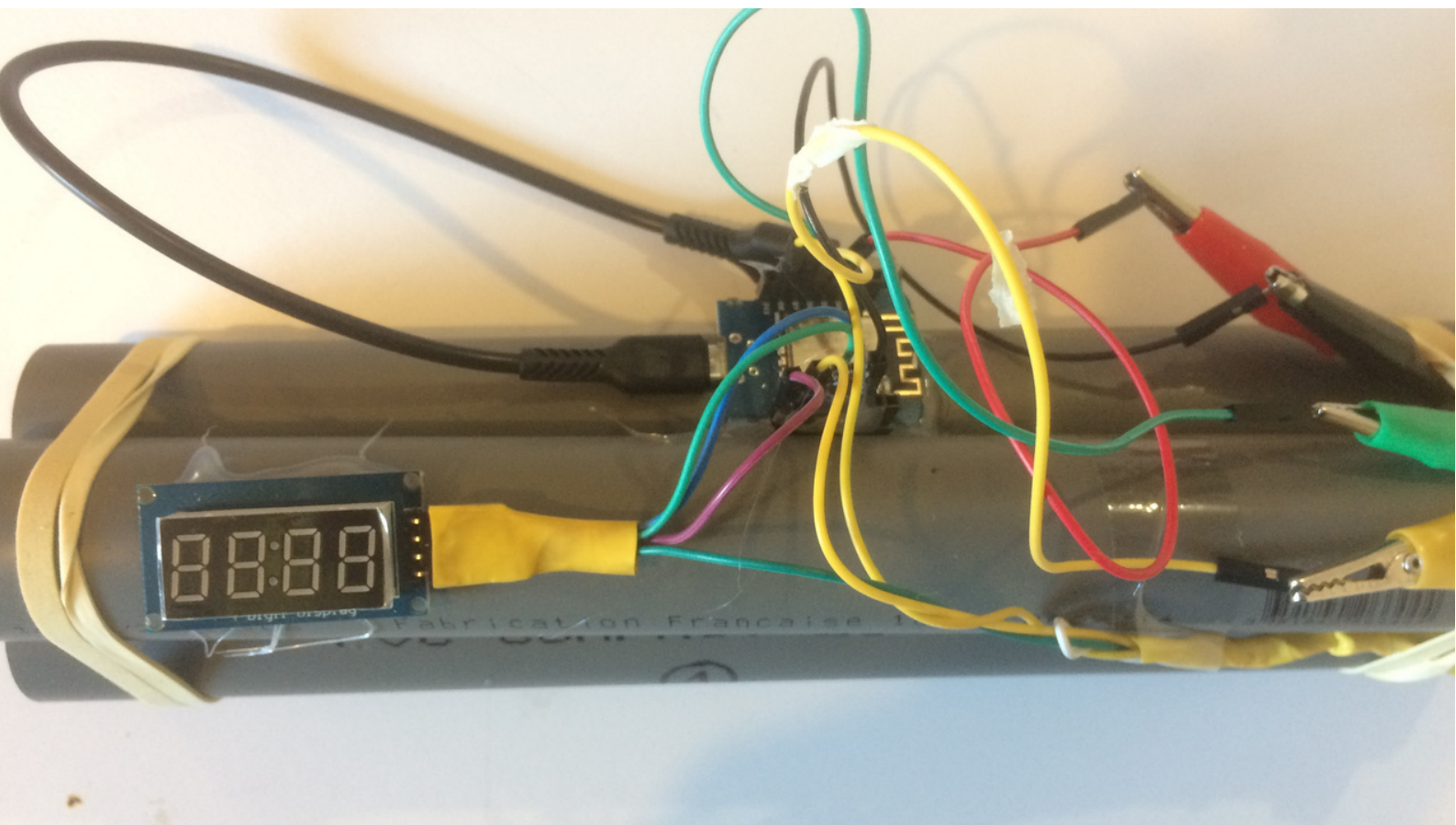
TIMER BOMB MATERIAL NEEDED



- Wemos D1 mini
- Crocodile wires
- 4 digits lcd display
- Jumper wires
- A buzzer tone
- A pvc pipe which you can cut in three equal parts
- Glue gun

The Arduino file is available here : [Download](#)

If you have gathered everything you can jump to Step 2.

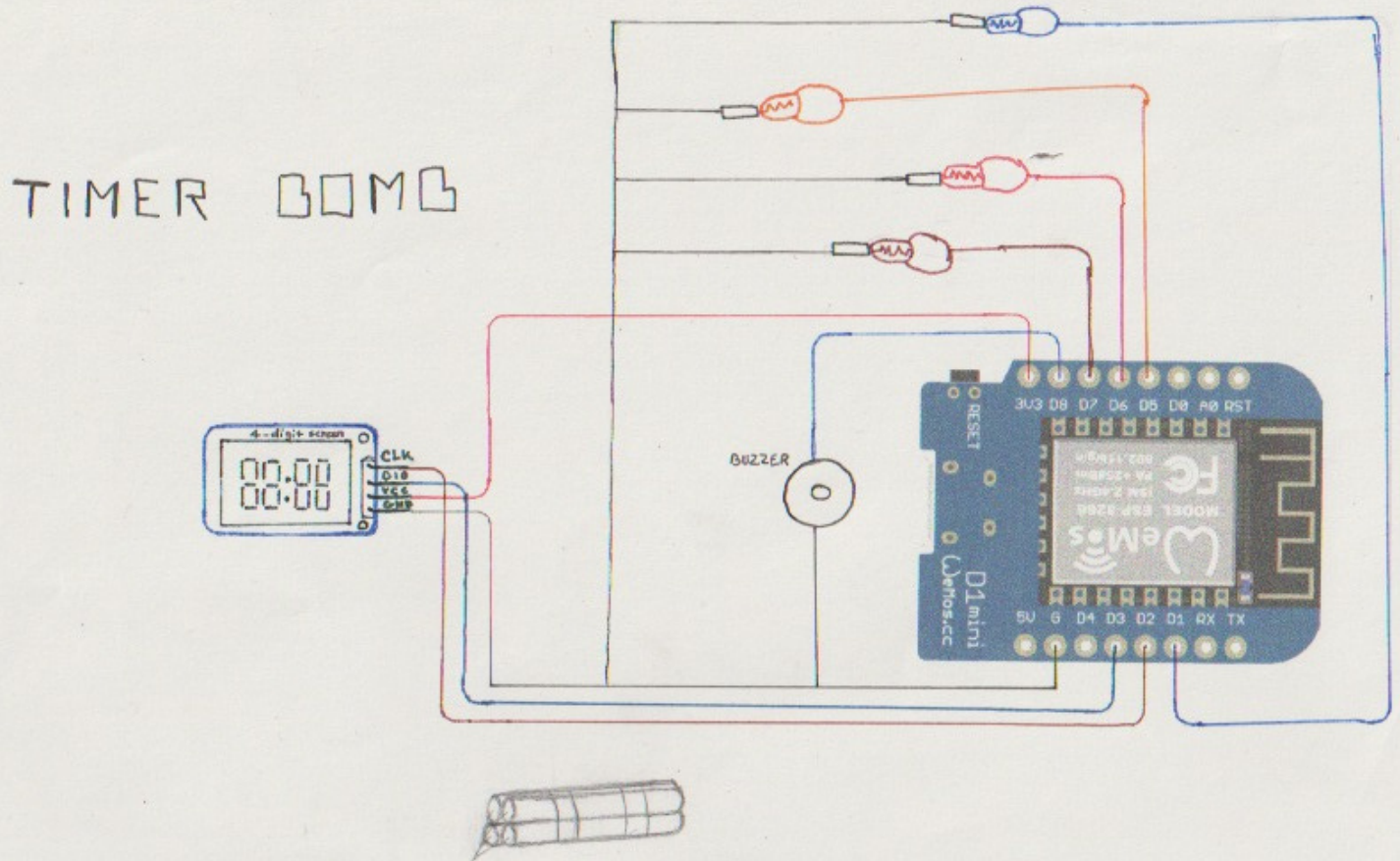


02

TIMER BOMB WIRING



Follow this scheme to make your prototype with a breadboard first.



03

TIMER BOMB UPLOADING THE CODE



Upload the code downloaded earlier to your Arduino Board and try out your prototype. The code is lightly complicated but you must understand that one part is for the music that the buzzer will play in case of failure and the other part is for the defusing process.

Inside this code you only have a few parameters you can change.

- Line 32 : It is the penalty time when a wrong wire is disconnected.
- Line 43 : The order of cutting the wire (0 meaning do not cut and 1 to 3 the order)
- Line 53 : The duration of the timer in minutes

```
cutthewires pitches.h
26
27 // CONSTANTS
28 const byte numWires = 4;
29 // Note that Wemos D1 mini has PULLDOWN resistor on D8, but no PULLUP
30 const int wirePins[numWires] = {D7, D6, D5, D1};
31 // Amount of time (in ms) to be deducted when an incorrect wire is cut
32 const unsigned long timePenalty = 600000;
33
34
35
36 // GLOBALS
37 // Create a display with the specified CLK/DIO lines
38 TM1637Display display(D2, D3);
39 int lastState[numWires];
40
41 // What is the order in which wires need to be cut
42 // 0 indicates the wire should not be cut!
43 int wiresToCut[numWires] = {0, 1, 3, 2}; //0, 1, 2, 3, so you need to cut 1 and then 3 and not 0 and 2
44 byte wiresCutCounter = 1;
45 // Keep track of the current state of the device
46 enum State {Inactive, Active, Defused, Exploded};
47 State state = State::Inactive;
48 // This is the timestamp at which the bomb will detonate
49 // It is calculated by adding on the specified number of minutes in the game time
50 // to the value of millis() when the code initialised.
51 unsigned long detonationTime;
52 // The game length (in minutes)
53 int gameDuration = 45;
54
55 void Activate(){
56     state = State::Active;
57     // Set the detonation time to the appropriate time in the future
58     detonationTime = millis() + (unsigned long)gameDuration*60*1000;
59     Serial.println(F("Bomb activated!"));
60 }
61
62 void Deactivate() {
63     Serial.println(F("Bomb defused"));
64     state = State::Inactive;
65 }
66
67 void Detonate() {
68     state = State::Exploded;
69     Serial.println("BOOM!");
70 }
```

04

TIMER BOMB MAKING THE BOMB



You're ready to assemble your bomb and finish with the prototype.

Cut your PVC pipe in three and glue the pieces together or tighten them with elastics.

You should glue the Arduino Board somewhere over the pipes once it seems stable. Do the same for the screen.

Connect all the crocodile clips and strap them together to make them stable.

If everything is connected you're ready to plug your bomb and see the timer start counting down!

