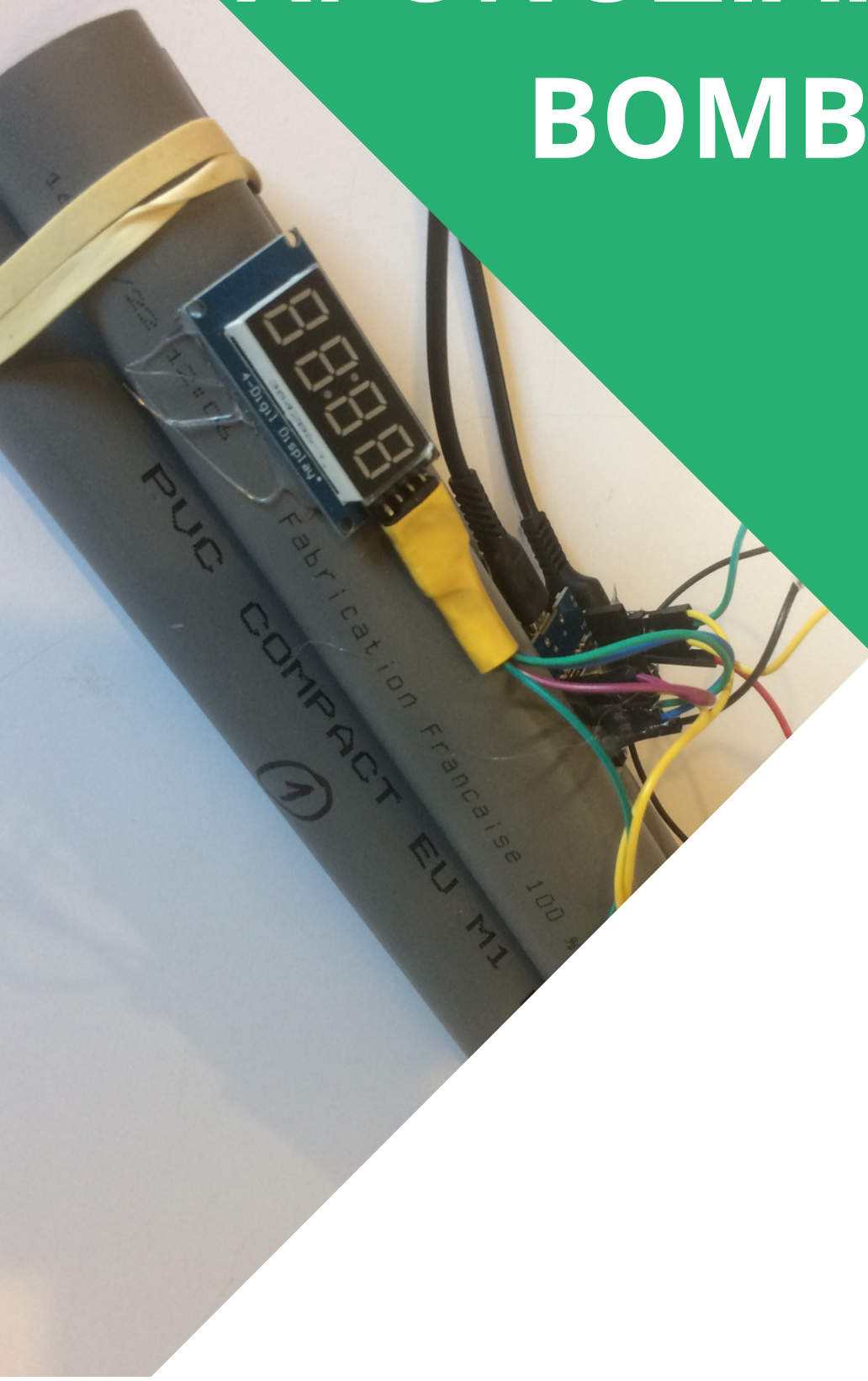




ΧΡΟΝΟΔΙΑΚΟΠΤΗΣ ΒΟΜΒΑΣ



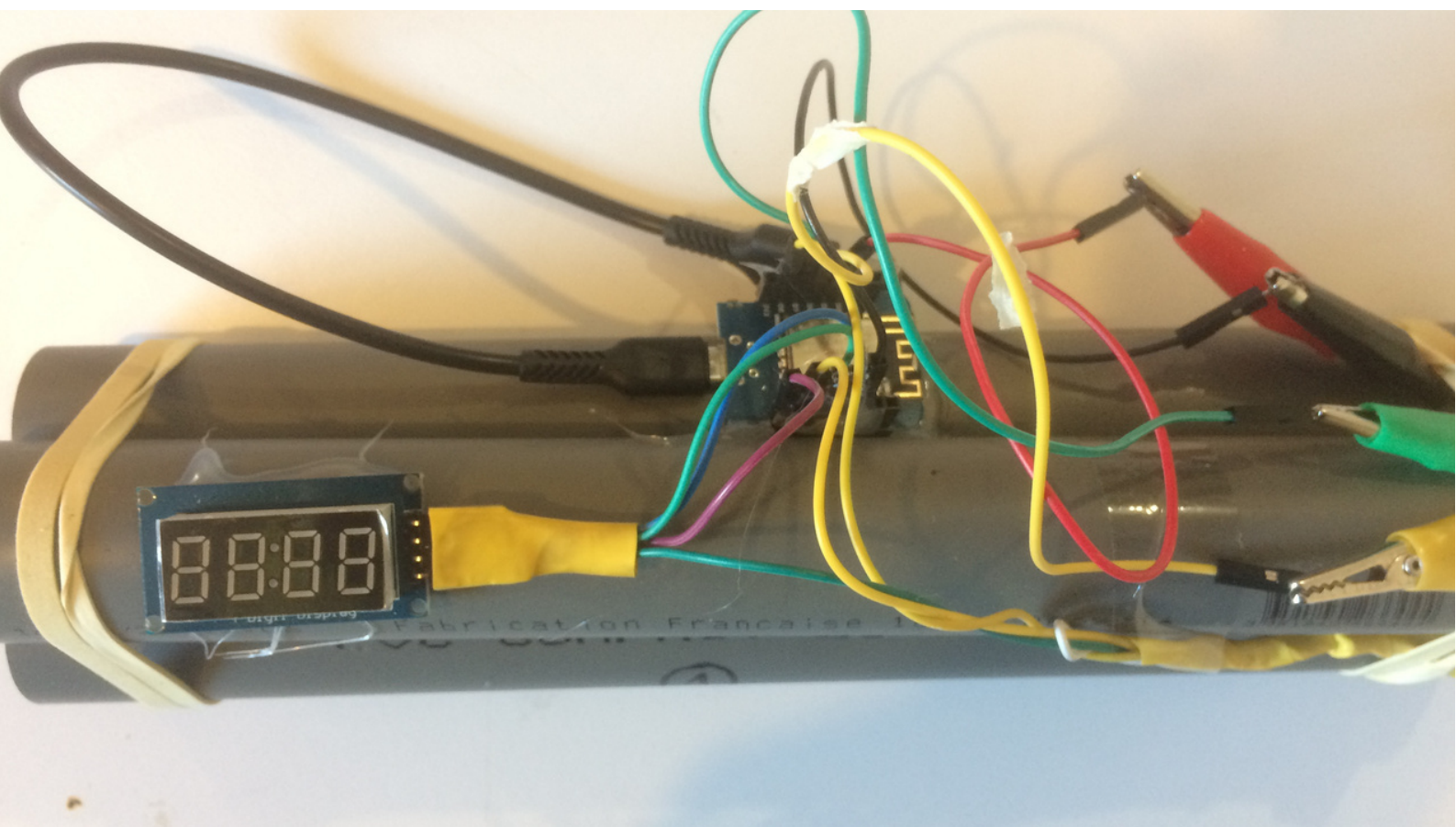
01 ΑΠΑΙΤΟΥΜΕΝΟ ΥΛΙΚΟ ΓΙΑ ΤΟΝ ΧΡΟΝΟΔΙΑΚΟΠΤΗ ΒΟΜΒΑΣ



- Wemos D1 mini
- Καλώδια με κροκοδειλάκια
- Οθόνη Icd 4 ψηφίων
- Καλώδια Jumper (καλώδια για πλακέτες δοκιμών breadboard)
- Ένας βομβητής για την εκπομπή ήχου
- Ένας σωλήνας rnc τον οποίο μπορείτε να κόψετε σε τρία ίσα μέρη
- Πιστόλι κολλητικής ταινίας

Το αρχείο του Arduino είναι διαθέσιμο εδώ : [Λήψη](#)

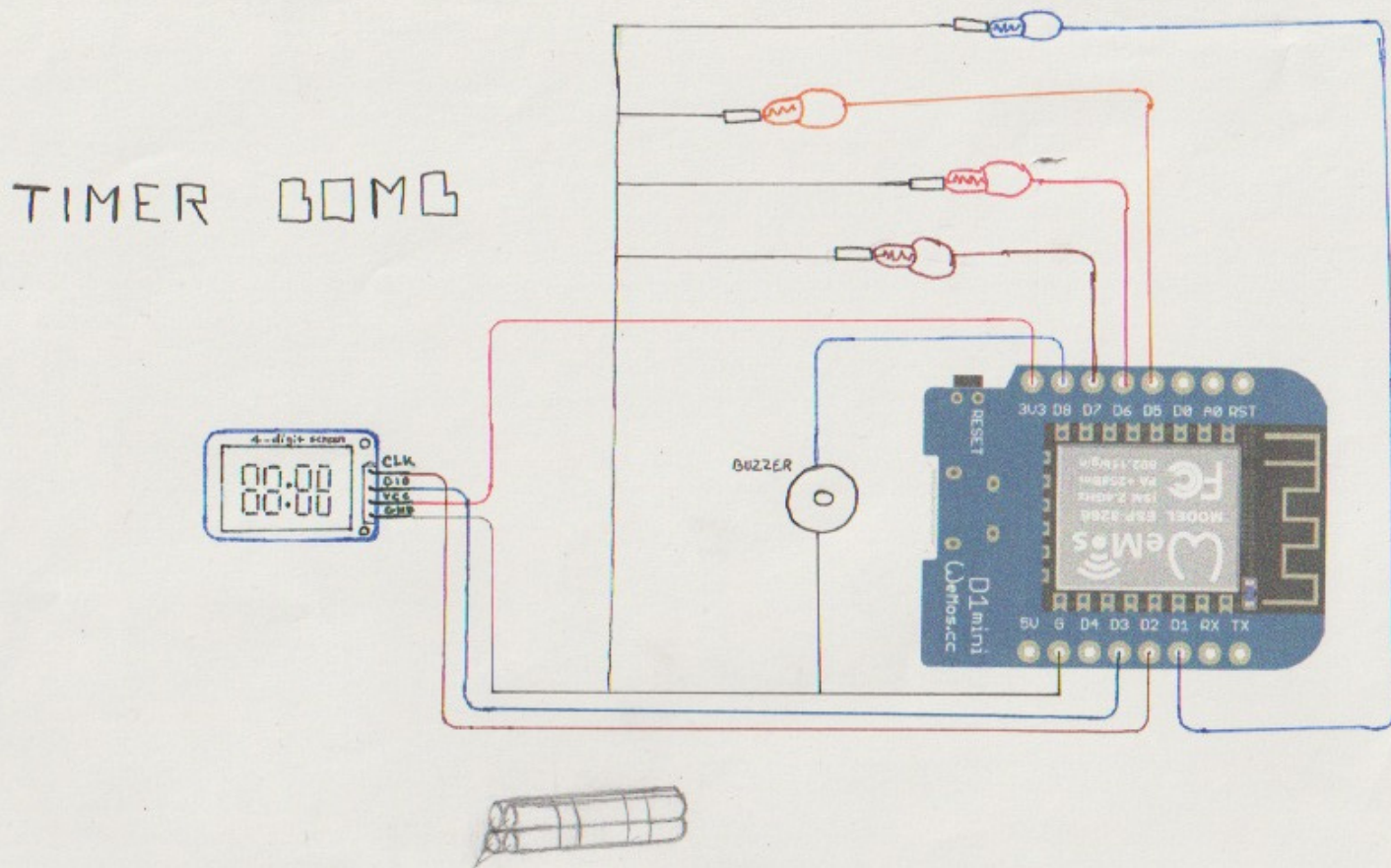
Αν έχετε συγκεντρώσει όλα τα υλικά, μπορείτε να μεταβείτε στο Βήμα 2.



02 ΚΑΛΩΔΙΩΣΗ ΤΟΥ ΧΡΟΝΟΔΙΑΚΟΠΤΗ ΒΟΜΒΑΣ



Ακολουθήστε το πιο κάτω σχέδιο για να φτιάξετε πρώτα το πρωτότυπό σας με την πλακέτα δοκιμών.



03 ΜΕΤΑΦΟΡΤΩΣΗ ΤΟΥ ΚΩΔΙΚΑ ΣΤΟΝ ΧΡΟΝΟΔΙΑΚΟΠΤΗ ΒΟΜΒΑΣ



Ανεβάστε τον κώδικα που κατεβάσατε νωρίτερα στην πλακέτα του Arduino και δοκιμάστε το πρωτότυπό σας. Ο κώδικας είναι ελαφρώς περίπλοκος, αλλά πρέπει να καταλάβετε ότι το ένα μέρος του κώδικα είναι για τη μουσική που θα παίζει ο βομβητής σε περίπτωση αποτυχίας και το άλλο μέρος είναι για τη διαδικασία απενεργοποίησης.

Μέσα σε αυτόν τον κώδικα έχετε μόνο λίγες παραμέτρους που μπορείτε να αλλάξετε.

- Γραμμή 32 : Ο χρόνος ποινής όταν αποσυνδέεται ένα λάθος καλώδιο.
- Γραμμή 43 : Η σειρά κοπής του καλωδίου (το 0 σημαίνει "να μην κοπεί" και το 1 μέχρι το 3 αναφέρεται στη σειρά κοπής του καλωδίου).
- Γραμμή 53 : Η διάρκεια του χρονοδιακόπτη σε λεπτά.

```
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 ΣΥΝΑΡΜΟΛΟΓΗΣΗ ΤΗΣ ΒΟΜΒΑΣ



Είστε έτοιμοι να συναρμολογήσετε τη βόμβα σας και να ολοκληρώσετε το πρωτότυπο.

Κόψτε τον σωλήνα PVC στα τρία και κολλήστε τα ξυλάκια μεταξύ τους ή σφίξτε τα με λάστιχα.

Αφου στεροποιηθούν, θα πρέπει πρώτα να κολλήσετε κάπου επάνω τους την πλακέτα Arduino. Κάντε το ίδιο και για την οθόνη.

Συνδέστε όλα τα κλιπ κροκοδειλάκια και δέστε τα επίσης μεταξύ τους για να τα σταθεροποιήσετε.

Αν όλα έχουν συναρμολογηθεί, τότε συνδέστε τη βόμβα με το χρονόμετρο για να αρχίσει να μετράει αντίστροφα!

