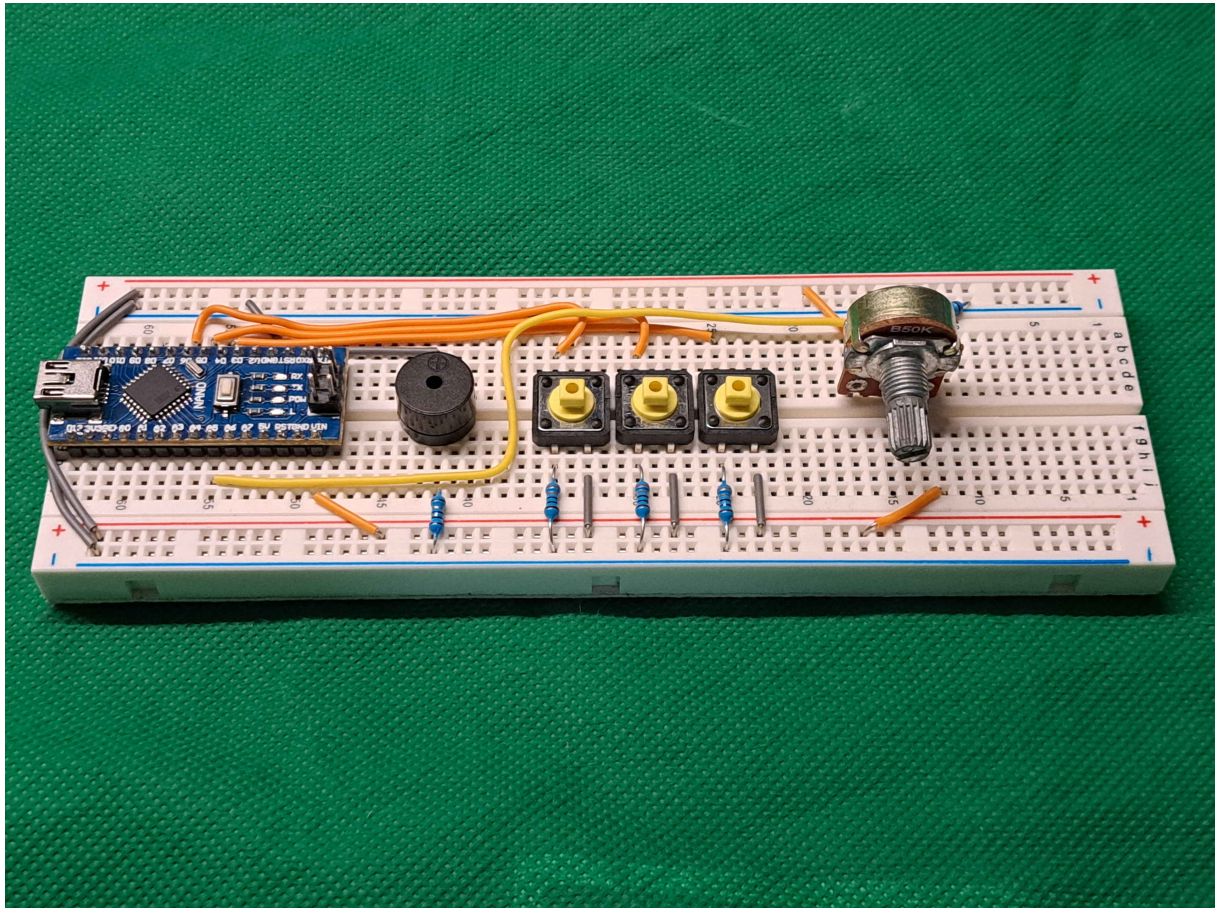


Solution: Clavier à Notes avec Buzzer passif



Code de l'Étape 1

```
1 #define PIN_BUZZER 11
2
3 void setup() {
4   // Configurer en sortie
5   pinMode(PIN_BUZZER, OUTPUT);
6
7   // Émettre un fréquence de 523 Hz
8   tone(PIN_BUZZER, 523);
9
10  // Attendre 1500ms
11  delay(1500);
12
13  // Stopper le son
14  noTone(PIN_BUZZER);
15 }
16
17 void loop() {}
```

Code de l'Étape 2

```
1  #define PIN_BUZZER 11
2
3  #define PIN_BTN_A 6
4  #define PIN_BTN_B 5
5  #define PIN_BTN_C 4
6
7  void setup()
8  {
9      pinMode(PIN_BUZZER, OUTPUT);
10
11     pinMode(PIN_BTN_A, INPUT);
12     pinMode(PIN_BTN_B, INPUT);
13     pinMode(PIN_BTN_C, INPUT);
14 }
15
16 void loop() {
17     if (digitalRead(PIN_BTN_A) == 1) {
18         tone(PIN_BUZZER, 523);
19     } else if (digitalRead(PIN_BTN_B) == 1) {
20         tone(PIN_BUZZER, 695);
21     } else if (digitalRead(PIN_BTN_C) == 1) {
22         tone(PIN_BUZZER, 783);
23     } else {
24         noTone(PIN_BUZZER);
25     }
26 }
```

Limitations: Les boutons vérifiées avant ont priorité sur les autres, une seule note peut être émise à la fois.

Code de l'Étape 3

```
1  #define PIN_BUZZER 11
2
3  #define PIN_BTN_A 6
4  #define PIN_BTN_B 5
5  #define PIN_BTN_C 4
6
7  #define PIN_CALIBRATE A0
8
9  int offset;
10
11 void setup()
12 {
13     pinMode(PIN_BUZZER, OUTPUT);
14
15     pinMode(PIN_BTN_A, INPUT);
16     pinMode(PIN_BTN_B, INPUT);
17     pinMode(PIN_BTN_C, INPUT);
18 }
19
20 void loop() {
21     offset = analogRead(PIN_CALIBRATE);
22
23     if (digitalRead(PIN_BTN_A) == 1) {
24         tone(PIN_BUZZER, 523 + offset);
25     } else if (digitalRead(PIN_BTN_B) == 1) {
26         tone(PIN_BUZZER, 695 + offset);
27     } else if (digitalRead(PIN_BTN_C) == 1) {
28         tone(PIN_BUZZER, 783 + offset);
29     } else {
30         noTone(PIN_BUZZER);
31     }
32 }
```