Serrure à code : contrôler l'accès

TP d'électronique et de programmation — 6 séances de 1 h · binôme · note /20

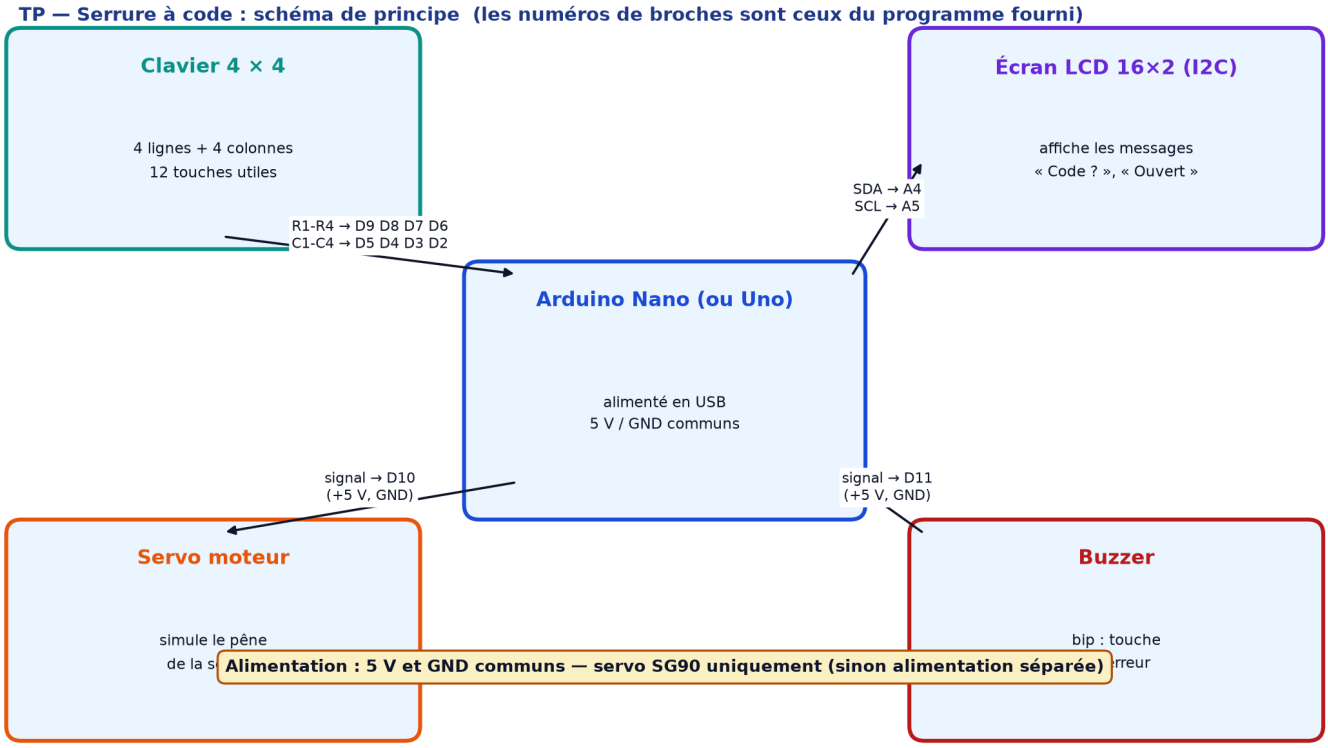
Pourquoi ce TP ? Une serrure à code, c'est déjà de la cybersécurité : on protège un accès par un **secret** (le code). Tu vas monter le système, le programmer, puis **l'attaquer** pour comprendre pourquoi un code à 4 chiffres ne suffit pas.

Ce que tu vas apprendre : câbler un clavier matriciel et un écran I2C · écrire un programme qui compare une saisie à un secret · piloter un servo · **chercher les faiblesses** d'un contrôle d'accès et proposer des corrections.

1. Le système et le matériel (par binôme)

Qté	Élément	Rôle	Remarque
1	**Arduino Nano** (ou Uno)	Lit le clavier, compare le code, commande le servo et l'écran	+ câble USB
1	**Clavier 4 × 4**	L'utilisateur tape le code (12 touches utiles sur 16)	nappe 8 fils
1	**Servo SG90**	Simule le pêne : 0° = fermé, 90° = ouvert	petit servo uniquement
1	**LCD 16×2 + module I2C**	Affiche « Code ? », « Ouvert », « Erreur », « Bloqué »	4 fils seulement
1	**Buzzer 5 V**	Bip court à chaque touche, bip long en cas d'erreur	actif, ou passif + `tone()`
1	**Plaque d'essai + fils**	Support du montage	~25 fils Dupont

3. Schéma de câblage



4. Tableau des connexions — à recopier et cocher au fur et à mesure

Module	Broche du module	Broche Arduino	Fait	Contrôle
Clavier	R1	D9	☐	appuie sur une touche : un bip doit se faire
	R2 / R3 / R4	D8 / D7 / D6	☐	
	C1 / C2	D5 / D4	☐	
	C3 / C4	D3 / D2	☐	
Servo	Signal (orange)	D10	☐	le servo tourne à 90° quand le code est bon
	+ (rouge)	5 V	☐	
	− (marron)	GND	☐	
LCD I2C	SDA	A4	☐	l'écran affiche « Code ? »
	SCL	A5	☐	
Alimentation	VCC LCD	5 V	☐	
	GND (tous)	GND	☐	
Buzzer	+ / −	D11 / GND	☐	bip à chaque touche

Sécurité : on travaille en 5 V par l'USB uniquement. On ne branche **jamais** ce montage sur une serrure réelle, ni sur une alimentation secteur. Le servo SG90 est alimenté par l'Arduino : au-delà, il faut une alimentation séparée, sinon l'Arduino redémarre.

5. Étape 1 — Tester le clavier (30 min)

Objectif : afficher sur le moniteur série la touche appuyée. Si rien ne s'affiche, le câblage du clavier est faux.

```
#include <Keypad.h>

const byte LIGNES = 4, COLONNES = 4;
char touches[4][4] = {{'1','2','3','A'},{'4','5','6','B'},
                      {'7','8','9','C'},{'*','0','#','D'}};
byte brochesLignes[LIGNES]     = {9, 8, 7, 6};
byte brochesColonnes[COLONNES] = {5, 4, 3, 2};

Keypad clavier = Keypad(makeKeymap(touches), brochesLignes, brochesColonnes, LIGNES, COLONNES);

void setup() {
  Serial.begin(9600);
  Serial.println("Test du clavier : appuie sur une touche");
}

void loop() {
  char t = clavier.getKey();
  if (t) {
    Serial.print("Touche : ");
    Serial.println(t);
  }
}
```

6. Étape 2 — Comparer le code et ouvrir (1 h 30)

Le code secret est `1234`. On tape 4 chiffres, puis `#` pour valider.

```
#include <Keypad.h>
#include <Servo.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);
Servo pene;
const String CODE_SECRET = "1234";
String saisie = "";
const byte BUZZER = 11;

void setup() {
  lcd.init(); lcd.backlight();
  pene.attach(10); pene.write(0);        // 0° = ferme
  pinMode(BUZZER, OUTPUT);
  lcd.print("Code ?");
}

void loop() {
  char t = clavier.getKey();
  if (!t) return;

  bip(80);
  if (t == '#') {                        // validation
```

```
    if (saisie == CODE_SECRET) {
      lcd.clear(); lcd.print("Ouvert");
      pene.write(90);                  // 90° = ouvert
      delay(3000);
      pene.write(0);
      lcd.clear(); lcd.print("Code ?");
    } else {
      lcd.clear(); lcd.print("Erreur");
      bip(600);
      delay(1500);
      lcd.clear(); lcd.print("Code ?");
    }
    saisie = "";
  } else if (t == '*') {               // effacer
    saisie = "";
    lcd.clear(); lcd.print("Code ?");
  } else {
    saisie += t;
    lcd.setCursor(0, 1);
    lcd.print("****");                 // on masque la saisie
  }
}

void bip(int duree) {
  digitalWrite(BUZZER, HIGH); delay(duree); digitalWrite(BUZZER, LOW);
}
```

À vérifier avant d'aller plus loin : le code bon ouvre, un mauvais code affiche « Erreur », la touche * efface, l'écran ne montre jamais les chiffres tapés (seulement des *).

7. Étape 3 — L'attaque : pourquoi ce système est faible

Ton binôme joue le rôle de l'attaquant. Chronomètre en main, essayez de trouver le code.

Attaque	Comment	Temps trouvé
Essais successifs	taper 0000, 0001, 0002… sans limite	
Observation	regarder les traces de doigts sur les touches	
Code par défaut	essayer 1234, 0000, 1111	
Écoute	le buzzer confirme chaque appui : on entend le nombre de chiffres	

Les 3 faiblesses à corriger

1. **Aucune limite d'essais** : avec 4 chiffres, il n'y a que 10 000 combinaisons — quelques heures suffisent.
2. **Code stocké en clair** dans le programme : toute personne qui lit le code source connaît le code.
3. **Pas de trace** : on ne sait pas qui a essayé, ni quand.

À faire (bonus, 2 points) : ajouter un **compteur d'essais** — au bout de 3 erreurs, l'écran affiche « Bloque 30s » et le clavier est ignoré pendant 30 secondes. Écris les lignes à ajouter dans `loop()`.

8. Questions (8 points)

1. Combien de combinaisons possibles avec un code à 4 chiffres ? Et avec 6 chiffres ? (1 pt)
2. Pourquoi affiche-t-on des * au lieu des chiffres ? (1 pt)
3. Que se passe-t-il si deux personnes connaissent le code ? Cite une solution. (1,5 pt)
4. Explique en 3 lignes pourquoi il ne faut pas écrire le code en clair dans un programme. (1,5 pt)
5. Le système fonctionne sans internet : est-ce un avantage ou un inconvénient pour la sécurité ? (1,5 pt)
6. Propose une amélioration qui rende l'attaque par essais successifs beaucoup plus longue. (1,5 pt)

9. Grille d'évaluation (/20)

Critère	Points	Observations
Câblage correct et vérifié (tableau coché)	4	
Clavier fonctionnel (étape 1)	2	
Code bon → ouverture / code faux → erreur	4	
Écran et buzzer cohérents	2	
Questions	6	
Bonus : limitation des essais	2	
Total	**/20**	

TP inspiré d'un projet publié sur how2electronics.com, réécrit pour le Bac Pro CIEL : câblage en 5 V par USB, programme simplifié, partie « attaque » ajoutée. Ne jamais brancher ce montage sur une serrure réelle.