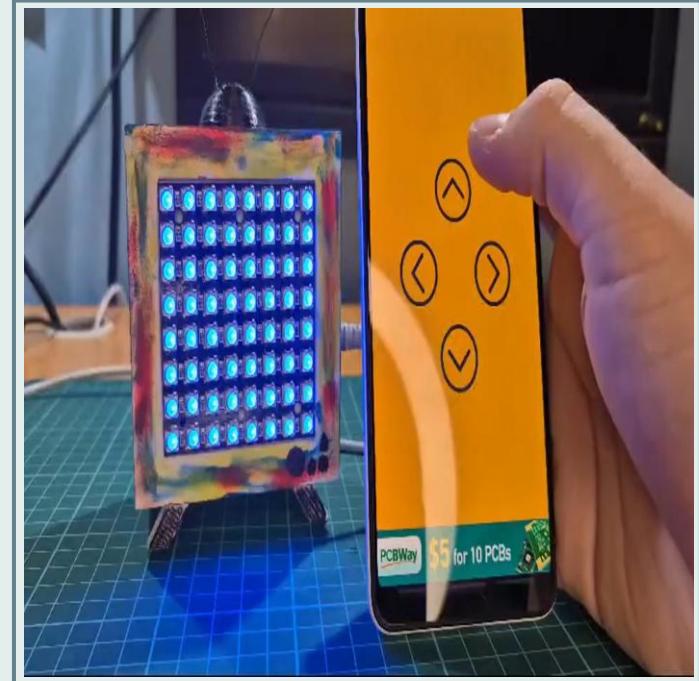


Projet Retro-Gaming Arduino



Le projet "**Retro-Gaming Arduino**" allie technologie moderne et esthétique nostalgique pour recréer l'expérience des jeux vidéo classiques. **Inspirée des télévisions portables des années 70-80, cette console propose une solution à la fois ludique et éducative.**

L'objectif est de créer une console portable fonctionnelle, au design personnalisable, qui combine apprentissage technique et plaisir.



Matériel, Logiciel et Design Utilisés

Matériel utilisé (Hardware)...

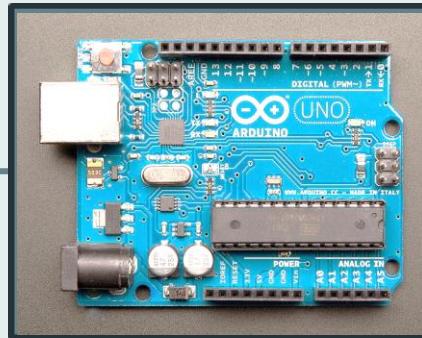
01



Matrice Neopixel 8x8

afficher des animations et des données de manière visuelle grâce à ses LED RGB.

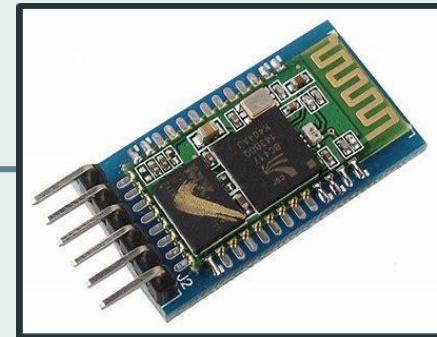
02



Carte Arduino Uno R3

Nous avons choisi cette carte pour sa simplicité, son coût abordable, sa large communauté d'utilisateurs et sa compatibilité avec divers modules.

03

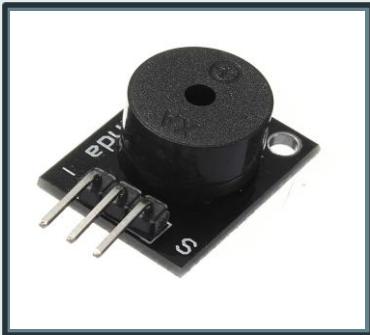


Module Bluetooth HC-05

Ce module a été intégré pour permettre une communication sans fil entre le système embarqué et une application smartphone.

...Matériel utilisé (Hardware)

04



Buzzer passif

05



Câbles Dupont

mettre des sons simples pour signaler
des événements ou des actions.

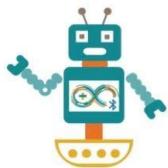
Logiciel utilisé (Software)



- PC avec l'IDE Arduino installé

fritzing

- PC avec FRITZIN installé



**Download Arduino
Bluetooth Control
APK for Android**

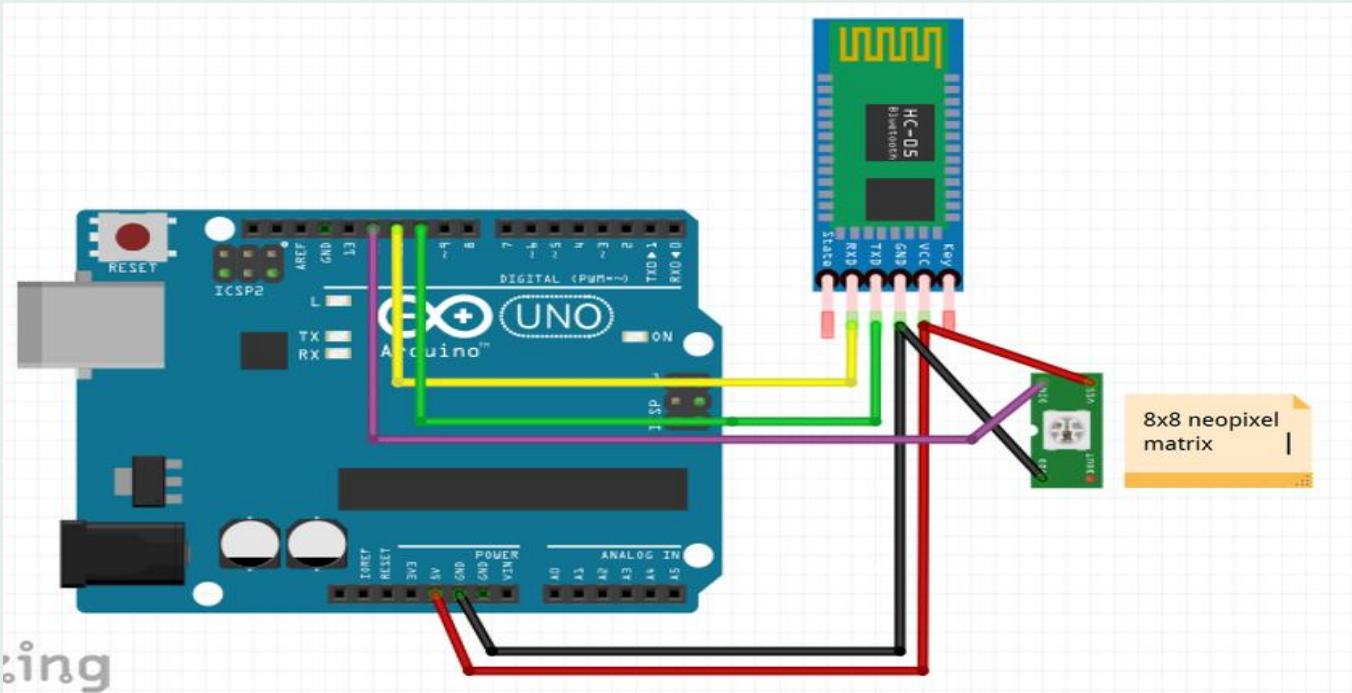
- Telephone avec Bluetooth installé

pour envoyer les commandes au système via le module Bluetooth.

03

Architecture

Architecture Matérielle



"Schéma de câblage pour un système interactif basé sur Arduino, Bluetooth HC-05, et une matrice LED Neopixel 8x8"

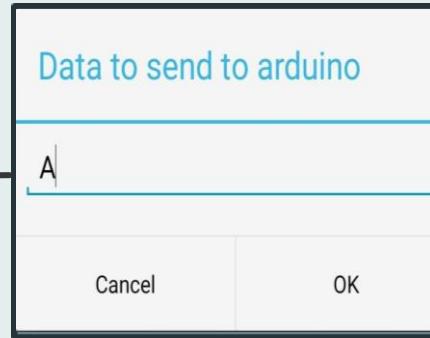
Architecture Logiciel

01



On appuie sur l'une des flèches directionnelles (haut, bas, gauche, droite).

02



L'application envoie un caractère spécifique au programme via une communication Bluetooth.

03

```
if (!jeuEnCours) {  
    if (monBT.available()) {  
        char donnee = monBT.read();  
        if (donnee == 'A') {  
            introductionMontree = false;  
            initialiserJeu();  
        }  
    }  
}  
return;
```

Commande 'A' : Réinitialisation du jeu

Architecture Logiciel

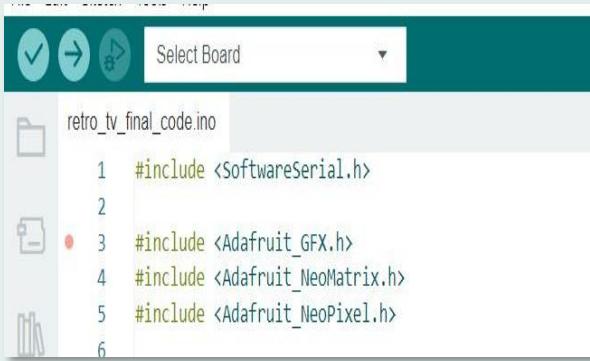
```
void setup() {  
    matrice.begin();  
    matrice.setTextWrap(false);  
    matrice.setBrightness(50);  
    matrice.setTextSize(1);  
  
    pinMode(BROCHE_BUZZER, OUTPUT);  
  
    // Initialisation de la communication Bluetooth  
    monBT.begin(38400);  
}
```

- ✖ **Initialisation du Système :**
Les paramètres de démarrage pour la matrice et les autres composants.

```
void loop() {  
    if (miBT.available()) {  
        char command = miBT.read();  
        switch (command) {  
            case 'L': // Allumer une animation LED  
                lightUpMatrix();  
                break;  
            case 'B': // Jouer un son avec le buzzer  
                playBuzzer();  
                break;  
        }  
    }  
}
```

La logique principale pour traiter les commandes Bluetooth

Architecture Logiciel



The screenshot shows the Arduino IDE interface with the file 'retro_tv_final_code.ino' selected. The code includes definitions for SoftwareSerial, Adafruit_GFX, Adafruit_NeoMatrix, and Adafruit_NeoPixel libraries. It also defines pin numbers for a matrix and a buzzer.

```
#include <SoftwareSerial.h>
#include <Adafruit_GFX.h>
#include <Adafruit_NeoMatrix.h>
#include <Adafruit_NeoPixel.h>
```

Inclusion des bibliothèques et configuration initiale

```
void lightUpMatrix() {
    for (int i = 0; i < 64; i++) {
        matriz.setPixelColor(i, matriz.Color(255, 0, 0)); // Rouge
    }
    matriz.show();
}
```

Animation LED : Une animation simple pour allumer toutes les LEDs en rouge.

```
#define BROCHE_MATRICE 12
#define BROCHE_HAUT 3
#define BROCHE_DROITE 4
#define BROCHE_BAS 5
#define BROCHE_GAUCHE 6
#define BROCHE_BUZZER 2
```

```
#define LARGEUR_MATRICE 8
#define HAUTEUR_MATRICE 8
#define LARGEUR_RAQUETTE 3
#define HAUTEUR_RAQUETTE 1
#define TAILLE_BALLE 1
```

Définition des broches et des paramètres pour chaque composant

```
void playBuzzer() {
    digitalWrite(PIN_BUZZER, HIGH);
    delay(200);
    digitalWrite(PIN_BUZZER, LOW);
}
```

Activation du Buzzer : Une fonction simple pour jouer un son. — —