



ECOLE NATIONALE SUPÉRIEURE D'ARTS ET MÉTIERS CASABLANCA

PROJET D'ÉLECTRONIQUE ANALOGIQUE RAPPORT

12V RGB LED Strip Controller

Réalisé par :

CHOUBIK Houssam
EL MAAROUFI Soukaina
LABBAD Mariam
MOUZAKKI Othmane

Encadré par :

Pr BAGHOURI Mostafa

Abstract

Ce projet a pour objectif la conception et la réalisation d'un contrôleur pour une bande LED (Light Emitting Diode) RGB alimentée en 12V, permettant un contrôle précis et personnalisable des couleurs et des animations lumineuses. Utilisant un microcontrôleur (Arduino , Esp ou équivalent) et la modulation par largeur d'impulsion (PWM), ce système offre une gestion fluide des transitions colorées tout en optimisant la consommation énergétique.

Le matériel inclut des transistors NPN pour la commande des canaux RGB, une alimentation adaptée pour garantir la stabilité du circuit, et une interface utilisateur simple permettant de modifier les couleurs et les modes d'éclairage.

Ce projet constitue une solution efficace pour des applications dans l'IOT (Internet of Things ou Internet des Objets), la domotique, la décoration événementielle ou les environnements commerciaux.

Les perspectives incluent l'intégration de fonctionnalités connectées via un module Wi-Fi , ainsi qu'un serveur web pour un contrôle à distance.

Abstract	1
1 Introduction	3
2 Étude théorique	3
2.1 Transistor TIP120	3
2.1.1 Caractéristiques du transistor TIP120	4
2.1.2 Avantages et limitations	5
2.1.3 Applications	6
2.1.4 Tests du transistor TIP120	6
2.2 Résistor	7
2.2.1 Code couleur du résistor	8
2.2.2 Avantages et désavantages du résistor	9
2.2.3 Mesurer une résistance : Mesurer une résistance avec un multimètre numérique	9
2.2.4 Les types des résistors	11
2.3 RGB LED Strip 12V	11
2.3.1 Structure d'une bande de LED RGB 12V	12
2.3.2 Structure d'une bande de LED RGB 12V	12
2.3.3 Alimentation et Contrôle	12
2.3.4 Avantages des bandes de LED RGB 12V	13
2.3.5 Applications courantes	13
2.4 Esp32	13
2.4.1 Caractéristiques techniques	14
2.4.2 Outils de développement et système	15
2.4.3 Applications	15
3 Conception et développement	16
3.1 Composants matériels	16
3.1.1 Esp32	16
3.1.2 Transistor TIP120	16
3.1.3 Résistor 10kΩ	17
3.1.4 Battery 9V	18
3.1.5 BreadBoard	18
3.1.6 RGB LED Strip 12V	19
3.1.7 fils	19
3.1.8 Carte de circuits imprimés	19
3.2 Théorie de fonctionnement	20
3.3 Circuit	20
3.4 Code pour programmer l'ESP32	21
3.5 Montage en utilisant la "BreadBoard"	24
3.6 La démonstration	24
4 Réalisation et Prototype	27
4.1 Réalisation	27
4.2 Prototype final	28
5 Conclusion	28
A Annexe :	29

1 Introduction

Dans le cadre du module d'électronique analogique, ce projet vise à mettre en pratique les concepts fondamentaux étudiés, notamment le fonctionnement des diodes, des transistors et leur rôle dans les circuits électroniques. L'éclairage LED RGB, technologie omniprésente dans les domaines de l'internet des objets et la domotique, constitue une application idéale pour illustrer ces principes.

Les bandes LED RGB, alimentées en 12V, permettent de produire une infinité de combinaisons de couleurs grâce au mélange des trois composantes primaires : rouge, vert et bleu. Toutefois, pour exploiter pleinement leur potentiel, un contrôleur est nécessaire. Celui-ci doit gérer la luminosité et les couleurs de manière précise.

Ce projet consiste à concevoir et réaliser un contrôleur de bande LED RGB basé sur l'utilisation de transistors NPN pour piloter les canaux RGB. Ces transistors, associés à une modulation par largeur d'impulsion (PWM), permettent de contrôler l'intensité des LED de manière fluide et précise. L'ensemble du système repose sur un microcontrôleur pour générer les signaux de commande, accompagné d'une interface pour permettre le choix de l'activation d'une ou de plusieurs couleurs.

Ce rapport présente les étapes clés du projet : de l'étude théorique à la conception, en passant par la réalisation matérielle et logicielle. Il met également en évidence les défis rencontrés et les solutions adoptées, tout en proposant des perspectives d'amélioration et des applications futures.

2 Étude théorique

Dans le cadre de ce projet, nous aborderons des composants électroniques clés qui vont nous permettre de développer notre projet. La compréhension des caractéristiques et du fonctionnement de ces composants est essentielle pour concevoir et implémenter des circuits électroniques performants.

2.1 Transistor TIP120

Le TIP120 est un transistor de puissance de type Darlington NPN largement utilisé dans les applications électroniques nécessitant un gain élevé en courant et une capacité à gérer des charges de puissance importante. Grâce à sa conception intégrant deux transistors en configuration Darlington, le TIP120 offre un gain en courant très élevé, ce qui le rend idéal pour des applications de commande de charges telles que les moteurs, les lampes ou les LED haute puissance.

Ce composant est encapsulé dans un boîtier robuste TO-220, qui facilite son montage sur dissipateur thermique pour gérer efficacement la chaleur générée lors de son fonctionnement. Le TIP120 est équipé de trois broches principales (Base, Collecteur, Émetteur) et inclut une diode interne pour protéger le circuit contre les surtensions causées par des charges inductives.

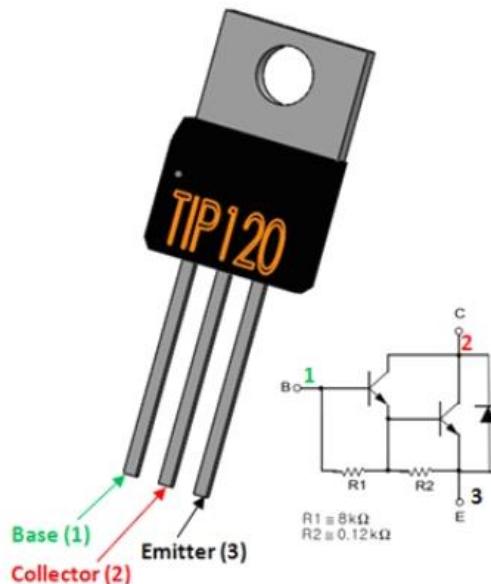


FIGURE 1 – Transistor TIP120

2.1.1 Caractéristiques du transistor TIP120

1. Caractéristiques électriques :

Le TIP120 présente des caractéristiques électriques impressionnantes qui le rendent adapté à des applications de puissance et de commutation. Voici quelques caractéristiques électriques de ce transistor :

- Type de transistor : NPN Darlington
- Tension maximale collecteur-émetteur (VCEO) : 60 V
- Tension maximale collecteur-base (VCBO) : 100 V
- Tension maximale émetteur-base (VEBO) : 5 V
- Courant collecteur maximal (IC max) : 5 A
- Courant collecteur de crête (IC peak) : 8 A (pour des courtes durées)
- Puissance maximale dissipée (Ptot) : 65 W (avec un dissipateur thermique approprié)
- Gain en courant (hFE) : 1000 à 5000 (typique à $IC = 3$ A)
- Tension de saturation collecteur-émetteur (VCE (sat)) : Environ 2 V (à $IC = 3$ A, $IB = 12$ mA)

2. Caractéristiques thermiques :

Les caractéristiques thermiques du TIP120 sont essentielles pour garantir une dissipation de chaleur efficace et un fonctionnement fiable dans des conditions de forte puissance. Voici quelques caractéristiques thermiques de ce transistor :

- Température de jonction maximale (T_j max) : 150 °C
- Résistance thermique jonction-boîtier ($R_{\theta JC}$) : 1,92 °C/W

3. Caractéristiques physiques :

Les caractéristiques physiques du TIP120, telles que son boîtier TO-220, permettent une installation facile et une gestion thermique optimale dans les circuits de puissance. Voici quelques caractéristiques physiques de ce transistor :

- Boîtier : TO-220 (boîtier plastique avec surface métallique pour dissipation thermique)
- Nombre de broches : 3 (Base, Collecteur, Émetteur)

4. Autres caractéristiques :

- Configuration Darlington : Intègre deux transistors NPN pour un gain de courant élevé
- Diode de protection intégrée : Diode de roue libre pour protéger contre les surtensions générées par des charges inductives.

2.1.2 Avantages et limitations

1. Avantages

- Gain de courant élevé : Grâce à sa configuration Darlington, le TIP120 offre un gain en courant (hFE) très élevé, généralement compris entre 1000 et 5000, ce qui permet de contrôler des charges importantes avec un faible courant de base.
- Capacité à gérer des charges élevées : Il peut supporter jusqu'à 5 A de courant collecteur et des tensions allant jusqu'à 60 V, ce qui le rend idéal pour des applications de puissance.
- Diode de protection intégrée : Une diode interne protège contre les surtensions générées par des charges inductives, comme des moteurs ou des solénoïdes.
- Boîtier robuste (TO-220) : Le boîtier TO-220, combiné avec une surface métallique, permet une dissipation thermique efficace et facilite le montage sur des dissipateurs de chaleur.
- Disponibilité et simplicité d'utilisation : Facile à trouver sur le marché, il est aussi simple à intégrer dans des projets éducatifs et professionnels.

2. Limitations

- Tension de saturation élevée : La tension de saturation collecteur-émetteur ($V_{CE(sat)}$) est relativement élevée (environ 2 V), ce qui entraîne une perte de puissance accrue dans certaines applications.
- Chauffe rapide : Sans un dissipateur thermique approprié, le TIP120 peut chauffer rapidement en raison de la puissance qu'il dissipe.
- Réponse lente : Comparé à des MOSFET modernes, sa configuration Darlington le rend moins rapide en commutation, ce qui peut limiter son utilisation dans des applications à haute fréquence.
- Courant de fuite élevé : En raison de la configuration Darlington, le courant de fuite de base est amplifié, ce qui peut poser des problèmes dans certains circuits sensibles.
- Remplacé par des alternatives modernes : Dans de nombreuses applications, des MOSFET ou d'autres transistors plus efficaces et rapides remplacent le TIP120 pour réduire la consommation d'énergie et améliorer les performances.

2.1.3 Applications

Le TIP120 est un transistor Darlington NPN polyvalent utilisé dans des circuits nécessitant une gestion de puissance et un gain élevé en courant. Voici ses principales applications :

1. Commande de moteurs électriques

Le TIP120 est idéal pour contrôler des moteurs à courant continu grâce à sa capacité à gérer des courants élevés et des charges inductives.

Exemples :

- Robots et systèmes automatisés.
- Commande de pompes ou ventilateurs via PWM (modulation de largeur d'impulsion).

2. Commande de LED et d'éclairage haute puissance

Grâce à sa robustesse et à sa gestion thermique, il est souvent utilisé pour piloter des LED haute puissance et des lampes :

Exemples :

- Dimmers (régulateurs de luminosité).
- Systèmes d'éclairage décoratif ou industriel.

3. Amplificateurs de puissance

Avec son gain élevé, il est utilisé pour amplifier des signaux afin de piloter des charges plus importantes.

Exemples :

- Amplificateurs audios.
- Circuits d'amplification pour capteurs ou signaux faibles.

4. Systèmes de protection et de régulation

Le TIP120 peut également servir dans des circuits de protection ou de régulation grâce à sa diode intégrée et à ses caractéristiques de puissance.

Exemples :

- Protection contre les surtensions dans des circuits inductifs.
- Stabilisation de tension dans des systèmes de régulation simple.

2.1.4 Tests du transistor TIP120

1. Tests de base avec un multimètre :

A L'aide d'utiliser un multimètre en mode "diode" pour tester les jonctions internes du transistor.

- **Étape 1 : Vérification de la jonction Base-Émetteur (B-E) :**
 - Placez la sonde positive du multimètre sur la base (B) et la sonde négative sur l'émetteur (E).
 - Une lecture typique se situe entre 0,6 V et 0,8 V.
 - Inversez les sondes : la lecture doit indiquer une valeur infinie (circuit ouvert).

- **Étape 2 : Vérification de la jonction Base-Collecteur (B-C) :**
 - Placez la sonde positive sur la base (B) et la sonde négative sur le collecteur (C).
 - Une lecture similaire entre 0,6 V et 0,8 V est attendue.
 - Inversez les sondes : la lecture doit être infinie.
- **Étape 3 : Vérification de la jonction Collecteur-Émetteur (C-E) :**
 - Placez la sonde positive sur le collecteur (C) et la sonde négative sur l'émetteur (E).
 - La lecture doit être infinie dans les deux sens.

2. Tests en circuit avec alimentation :

Pour vérifier le fonctionnement dans un circuit, procédez comme suit :

- **Étape 1 : Configuration simple de test :**
 - Connectez une résistance ($1 \text{ k}\Omega$) entre la base (B) et la source de commande (par exemple, une alimentation 5 V).
 - Connectez une LED avec une résistance en série entre le collecteur (C) et une alimentation externe (par exemple, 12 V).
 - Reliez l'émetteur (E) à la masse.
- **Étape 2 : Observation :**
 - Lorsque la base reçoit un signal (5 V via la résistance), la LED doit s'allumer, indiquant que le transistor commute correctement.
 - Si la LED reste éteinte ou toujours allumée, le transistor peut être défectueux.

2.2 Résistor

Un résistor, également appelé résistance, est un composant électronique essentiel utilisé dans les circuits électriques pour limiter le courant, diviser la tension ou fournir une charge spécifique. Les résistors fonctionnent en opposant une certaine résistance au passage du courant électrique, convertissant ainsi une partie de l'énergie électrique en chaleur. La valeur de résistance d'un résistor est mesurée en ohms (Ω) et est déterminée par la composition du matériau résistif et les dimensions du composant.

Les résistors peuvent être fabriqués à partir de divers matériaux, notamment du carbone, du métal ou des films métalliques, et sont disponibles sous différentes formes, telles que des résistors fixes, ajustables ou variables.



FIGURE 2 – Résistor

2.2.1 Code couleur du résistor

Les couleurs employées pour le marquage de valeurs des résistances, rangées par ordre croissant de valeur en partant de zéro sont : 0 = Noir, 1 = Brun ou Marron, 2 = Rouge, 3 = Orange, 4 = Jaune, 5 = Vert, 6 = Bleu, 7 = Violet, 8 = Gris, 9 = Blanc .

Couleur	Valeur	Multiplicateur
Noir	0	$\times 1$
Marron	1	$\times 10$
Rouge	2	$\times 100$
Orange	3	$\times 1000$
Jaune	4	$\times 10000$
Vert	5	$\times 100000$
Bleu	6	$\times 1000000$
Violet	7	$\times 10000000$
Gris	8	$\times 100000000$
Blanc	9	$\times 1000000000$

TABLE 1 – Tableau de Code Couleur pour la Résistance

Couleur	Couleur	1er anneau	2e anneau	3e anneau	4e anneau Multiplicateur	5e anneau Tolérance
Noir		0	0	0	$\times 1\Omega$	
Brun		1	1	1	$\times 10\Omega$	$\pm 1\%$
Rouge		2	2	2	$\times 100\Omega$	$\pm 2\%$
Orange		3	3	3	$\times 1k\Omega$	
Jaune		4	4	4	$\times 10k\Omega$	
Vert		5	5	5	$\times 100k\Omega$	$\pm 0.5\%$
Bleu		6	6	6	$\times 1M\Omega$	$\pm 0.25\%$
Violet		7	7	7	$\times 10M\Omega$	$\pm 0.10\%$
Gris		8	8	8	$\times 100M\Omega$	$\pm 0.05\%$
Blanc		9	9	9	$\times 1G\Omega$	
Or					$\times 0.1\Omega$	$\pm 5\%$
Argent					$\times 0.01\Omega$	$\pm 10\%$

FIGURE 3 – Code couleur pour la résistance

Exemple :

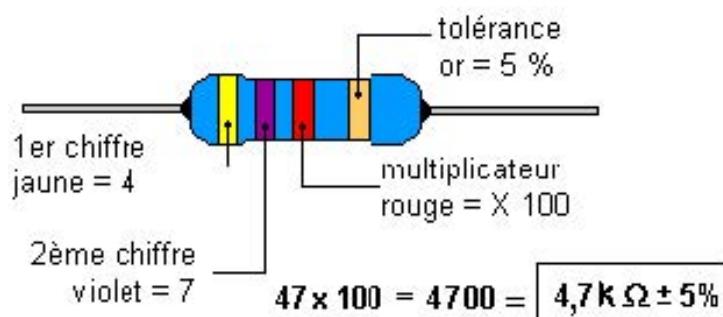


FIGURE 4 – Exemple d'une résistance

2.2.2 Avantages et désavantages du résistor

1. Les avantages

- Les résistances sont utilisées pour réduire le flux de courant, ajuster les niveaux de signal, diviser les tensions, polariser les éléments actifs et terminer les lignes de transmission.
- La connexion d'une résistance réduit le courant et protège les instruments contre les courants élevés. La résistance augmente également la durée de vie de la batterie car moins de courant est consommé.
- Les résistances MELF présentent des avantages tels que la résistance aux vibrations, aux chocs thermiques, une meilleure dissipation de la chaleur et une stabilité à long terme en fonctionnement. Par conséquent, elles sont largement utilisées dans les domaines du contrôle industriel et des applications Automotrice.

2. Les désavantages

- Les résistances à haute résistance s'opposent à une grande quantité de courant électrique. Par conséquent, une grande quantité d'énergie est gaspillée sous forme de chaleur.
- Les résistances sont sensibles aux variations de température, ce qui peut modifier leur valeur de résistance.
- Les résistances ont une puissance maximale qu'elles peuvent supporter (exprimée en watts). Si cette limite est dépassée, elles risquent de surchauffer ou de s'endommager.

2.2.3 Mesurer une résistance : Mesurer une résistance avec un multimètre numérique

— **Etape 1 : Choisissez un élément dont vous désirez connaître la résistance.**

Afin d'avoir la mesure la plus exacte possible, retirez l'élément du circuit sur lequel il est monté ou testez-le avant de le monter. En effet, on ne mesure pas une résistance quand le composant est sous tension, les autres éléments du circuit pouvant fausser la mesure à cause de leurs propres résistances.

Si vous testez la résistance d'un circuit ou si vous en démontez un élément, n'oubliez pas de couper le courant dans ledit circuit !

— **Etape 2 : Branchez les cordons dans les bonnes bornes.**

Les multimètres sont livrés avec deux cordons, l'un rouge, l'autre noir. Sur leurs façades, les multimètres présentent des bornes qui servent à mesurer la résistance, la tension ou l'intensité d'un courant. Pour la mesure de la résistance, seules deux bornes nous concernent : celle marquée « COM » (pour « commun ») et celle identifiée par la lettre grecque oméga, Ω (symbole de l'ohm). Le cordon noir est toujours introduit dans la borne marquée « COM », le rouge dans la borne Ω .

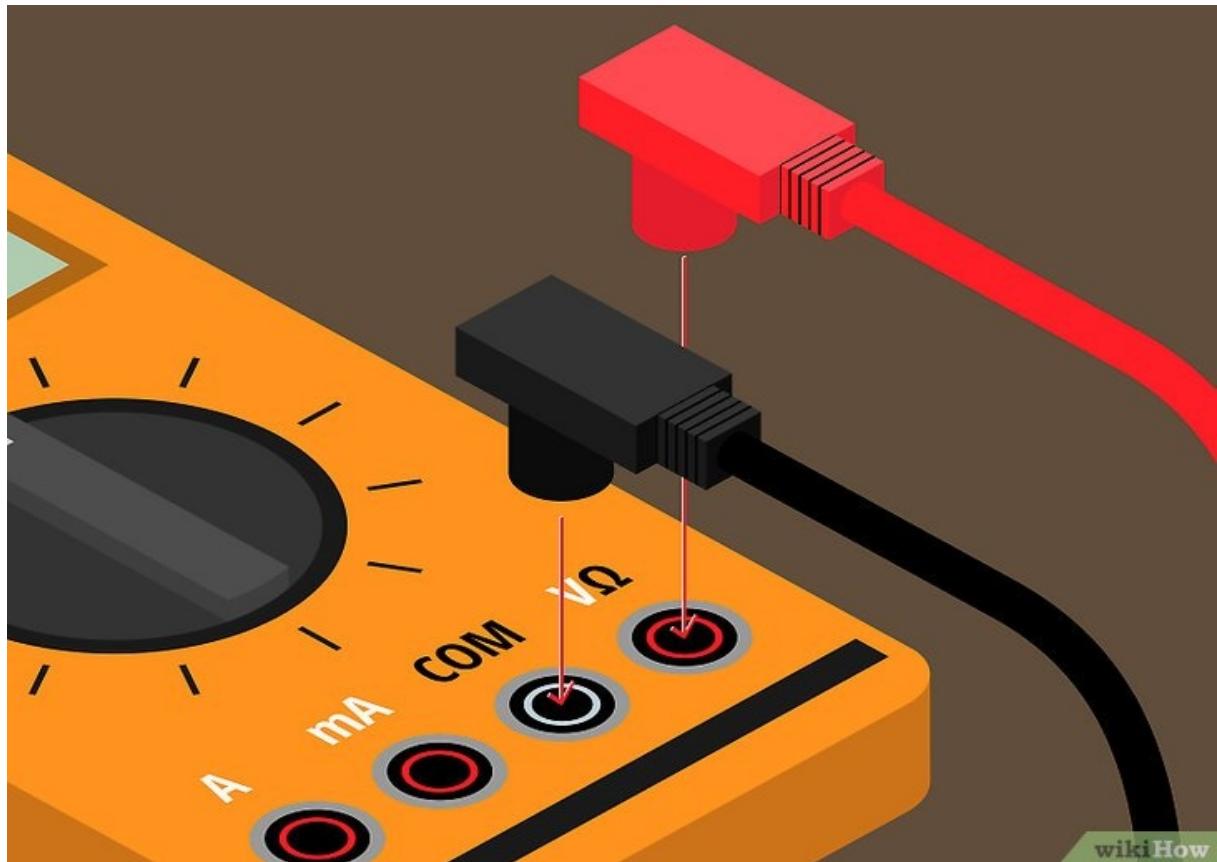


FIGURE 5 – Branchement du multimètre

— **Etape 3 : Mettez le multimètre en marche**

Choisissez ensuite le bon calibre. Une résistance peut aller de quelques fractions d'ohm ($0,1\text{ ohm}$) à plusieurs centaines de milliers ($1\ 000\ 000\text{ ohms} = 1\text{ M}\Omega$). Si vous voulez lire la bonne mesure, il faut choisir le bon calibre, c'est-à-dire l'intervalle de valeurs entre lesquelles s'inscrit la résistance mesurée.

— **Etape 4 : Positionnez bien les touches du multimètre.**

Comme vous l'avez fait lors du réglage, placez une des touches sur une des extrémités du composant et l'autre, sur l'autre extrémité. Maintenez la position jusqu'à ce que l'affichage devienne stable. Notez le résultat qui apparaît : c'est la résistance de votre composant.

2.2.4 Les types des résistors

On en trouve de nombreux types, différents par leur structure, leur forme, leurs caractéristiques électriques selon la technique de fabrication adoptée et l'emploi auquel elles sont destinées. On peut classer les résistances suivant le modèle de la figure 6 :

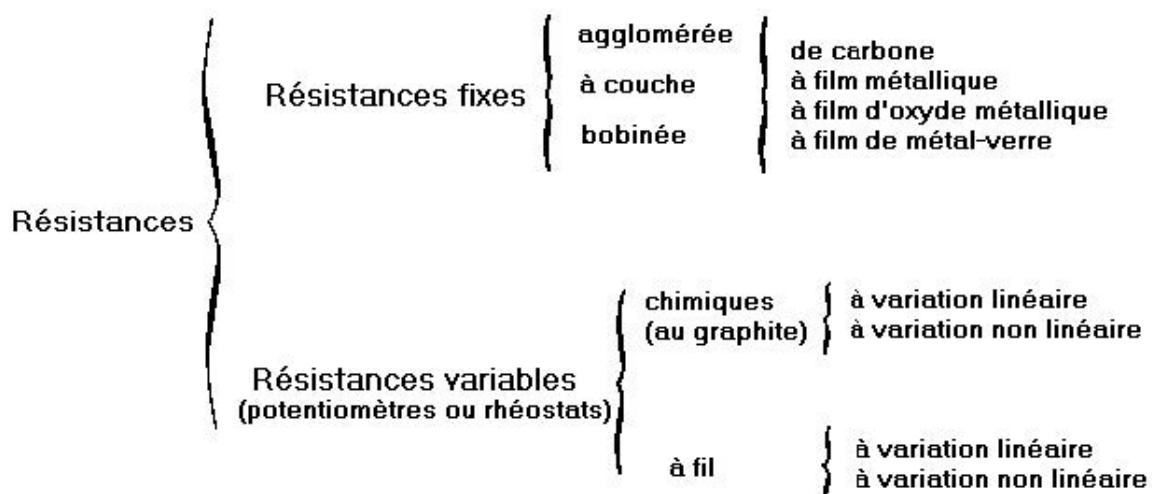


FIGURE 6 – Type des résistors

2.3 RGB LED Strip 12V

Les bandes de LED RGB 12V sont des composants utilisés dans l'éclairage décoratif et les projets d'automatisation. Elles offrent une flexibilité importante grâce à leur capacité à produire une large gamme de couleurs en combinant des LEDs rouges, vertes et bleues. Le contrôle de ces bandes permet de créer des effets lumineux dynamiques et variés.



FIGURE 7 – RGB LED Strip 12V

2.3.1 Structure d'une bande de LED RGB 12V

Une bande de LED RGB 12V est composée de plusieurs LEDs organisées sur une bande flexible. Chaque LED sur la bande est une diode électroluminescente (LED) capable de produire différentes couleurs en fonction des tensions appliquées sur les broches de chaque couleur (Rouge, Vert et Bleu). Les LEDs sont alimentées par une tension continue de 12V, ce qui en fait des éléments idéaux pour les applications nécessitant un éclairage puissant et une consommation énergétique modérée.

2.3.2 Structure d'une bande de LED RGB 12V

Les bandes RGB se composent de trois sous-LEDs dans chaque unité (une rouge, une verte et une bleue). En ajustant l'intensité de chaque sous-LED, on peut obtenir une variété de couleurs par mélange additif. Les couleurs primaires générées sont :

- **Rouge** : Activant uniquement la LED rouge.
- **Vert** : Activant uniquement la LED verte.
- **Bleu** : Activant uniquement la LED bleue.
- Couleurs mixtes : En combinant différentes intensités des trois LEDs, il est possible de produire des couleurs telles que le cyan, le magenta, le jaune, ou des nuances de blanc.

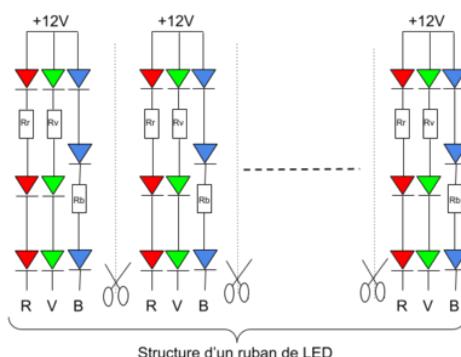


FIGURE 8 – Structue d'un ruban LED 12V

2.3.3 Alimentation et Contrôle

Les bandes de LED RGB 12V nécessitent une alimentation stable de 12V, et l'intensité de chaque couleur est souvent contrôlée à l'aide de PWM (modulation de largeur d'impulsion). Cela permet de réguler la quantité d'énergie envoyée à chaque LED et d'ajuster l'intensité lumineuse. Pour un contrôle plus précis et un changement de couleur dynamique, un contrôleur est souvent utilisé pour gérer l'intensité de chaque canal (R, G, B) et créer des effets comme le fondu, le clignotement ou des animations.

Les contrôleurs peuvent être analogiques ou numériques (en utilisant des microcontrôleurs comme l'ESP32, Arduino, ou autres). Certains modèles de bandes RGB sont équipés de puces intégrées qui permettent de programmer directement des effets sans nécessiter de contrôleur externe.

2.3.4 Avantages des bandes de LED RGB 12V

- **Flexibilité** : La bande est flexible et peut être découpée à des intervalles spécifiques pour s'adapter aux dimensions souhaitées.
- **Efficacité énergétique** : Les LEDs consomment moins d'énergie par rapport à d'autres sources lumineuses traditionnelles, comme les ampoules incandescentes.
- **Longévité** : Les LEDs ont une durée de vie bien plus longue que les autres technologies d'éclairage.
- **Contrôle de couleur** : Elles offrent une large gamme de couleurs grâce au contrôle individuel des LEDs rouges, vertes et bleues.

2.3.5 Applications courantes

Les bandes de LED RGB 12V sont utilisées dans diverses applications, telles que :

- **Éclairage décoratif** : Pour des effets lumineux dans les salons, les chambres ou les espaces commerciaux.
- **Éclairage extérieur** : Éclairage des jardins, des façades ou des terrains de sport.
- **Automatisation et IoT** : Grâce à leur intégration facile dans des systèmes de contrôle tels que les microcontrôleurs, elles sont largement utilisées dans des projets de domotique ou d'automatisation.

2.4 Esp32

L'ESP32 est une série de microcontrôleurs de type système sur une puce (SoC) d'Espressif Systems, basé sur l'architecture Xtensa LX6 de Tensilica (en), intégrant la gestion du Wi-Fi et du Bluetooth (jusqu'à LE 5.0 et 5.1) en mode double, et un DSP. C'est une évolution d'ESP8266. Le principal outil de développement est ESP-IDF, logiciel libre développé par Espressif, écrit en C et utilisant le système temps réel FreeRTOS. Il intègre un nombre important de bibliothèques et on retrouve dans son écosystème des bibliothèques tierce libres pour différents types de périphériques liés à l'embarqué et au temps réel. Le ESP32-C3, WiFi et BLE, est une variante, annoncée et sorti en novembre 2020, compatible broche à broche avec l'ESP8266, mais utilisant l'architecture RISC-V 32 bits plutôt que Xtensa . Le support du développement via la plateforme et l'IDE d'Arduino est disponible depuis la bibliothèque ESP32 2.0.0. Plus généralement, La série ESP32-C est basée sur RISC-V et la série ESP32-S sur Xtensa LX6. Son support Wi-Fi et Bluetooth, en fait un système apprécié dans le domaine de l'internet des objets. Ce SoC rencontre un certain succès depuis quelques années à la fois pour son coût, ses capacités et son intégration dans un nombre croissant de systèmes.

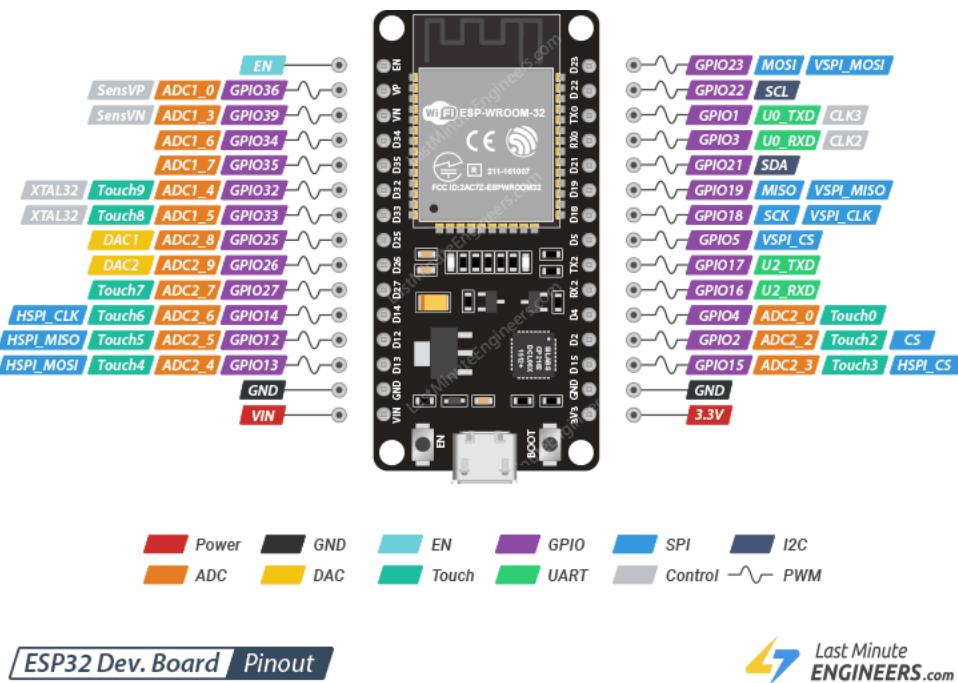


FIGURE 9 – ESP32

2.4.1 Caractéristiques techniques

L'ESP32 est un microcontrôleur très populaire qui est largement utilisé pour le développement de projets électroniques, notamment dans le domaine de l'Internet des objets (IoT). Voici quelques caractéristiques physiques courantes de la carte ESP32 :

- Processeur (CPU) : L'ESP32 est équipé d'un processeur à double cœur Tensilica Xtensa LX6, cadencé à des fréquences variables, généralement entre 80 MHz et 240 MHz.
- Mémoire : Il dispose généralement d'une mémoire flash intégrée pour le stockage du programme (jusqu'à plusieurs mégaoctets), ainsi que d'une mémoire RAM pour le stockage temporaire des données.
- Connectivité sans fil : L'ESP32 prend en charge différentes options de connectivité sans fil, notamment le Wi-Fi et le Bluetooth. Il peut être utilisé pour la communication sans fil et la connectivité réseau.
- GPIO (General Purpose Input/Output) : L'ESP32 est doté d'un certain nombre de broches GPIO qui peuvent être utilisées pour interagir avec d'autres composants électroniques, tels que des capteurs, des actionneurs etc.
- Interfaces : Il dispose de diverses interfaces, y compris des interfaces série (UART, SPI, I2C), des ports analogiques pour les capteurs analogiques, des interfaces pour les cartes SD, etc.
- Antenne : L'ESP32 intègre souvent une antenne intégrée pour le Wi-Fi et le Bluetooth, bien que certaines versions puissent avoir des connecteurs externes pour des antennes externes.
- Alimentation : L'ESP32 peut être alimenté via une source externe (par exemple, une batterie ou une alimentation électrique) et dispose généralement de régulateurs de tension intégrés pour fournir des tensions de fonctionnement stables.

- Dimensions : Les cartes ESP32 existent dans différentes tailles et formes, mais elles sont généralement compactes pour faciliter l'intégration dans des projets électroniques.
- Programmation : L'ESP32 peut être programmé à l'aide de l'IDE Arduino, PlatformIO, ou d'autres environnements de développement. Il peut également être programmé en utilisant le langage de programmation C/C++.

2.4.2 Outils de développement et système

L'Esp32 peut être supporté par les outils de programmation suivants :

- ESP-IDF (développement en C, avec outils en Python, se base sur FreeRTOS, c'est le système de développement de base).
- Apache NuttX, système compatible POSIX, porté sur ESP32-C3 (RISC-V) uniquement.
- Arduino IDE avec le module ESP32 Arduino Core, interface en Java, développement en C++.
- Espruino.
- ESPHome, spécialisé domotique, comportant une interface web pour l'interaction.
- FAUST, langage de programmation de traitement de données audio, utilisant son DSP.
- MicroPython, une variante pour l'embarqué du langage Python ; mruby, une variante pour l'embarqué du langage Ruby.
- NodeMCU.
- ESP32forth, programmation en Forth 32 bits pour ESP32.
- MicroEJ.

2.4.3 Applications

- **IOT ou Internet des objets** : Maisons connectés.
- **Synthétiseurs** : L'ESP32 est adapté à la création des Synthétiseurs avancés, dont des synthétiseurs analogiques. Le langage FAUST ou la bibliothèque ESP32soundsynth permettent de faciliter le développement de synthétiseurs basés sur cette architecture[23]. L'échantillonnage utilisant l'ADC intégré peut être effectué via la plate-forme de développement Arduino. Il est notamment adapté à des synthétiseurs au format Eurorack, comme le CTAG-Strämpler. Qun-synthesizer est un synthétiseur analogique portable, basé sur la carte ESP32-LyraT, fonctionnant avec la carte mère Nunomo.
- **Console de jeu portable** : Hardkernel, a créé en 2018 Odroid-Go, une console de jeu portable à très bas prix basée sur cette puce.

3 Conception et développement

3.1 Composants matériels

3.1.1 Esp32

Pour contrôler l'intensité des LEDs et permettre l'application de l'IOT (Internet des Objets).



FIGURE 10 – ESP32 DEVKITV1

3.1.2 Transistor TIP120

On aura besoin de 3 transistors TIP120.

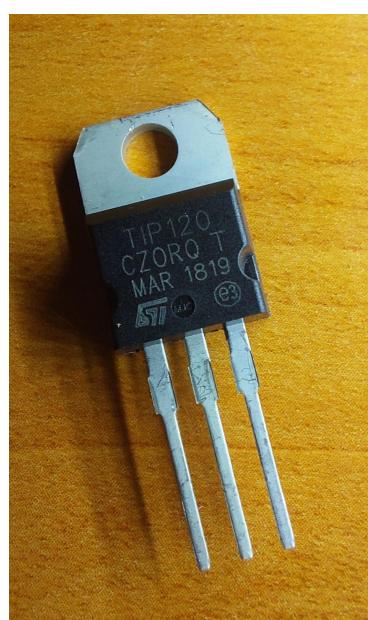


FIGURE 11 – Transistor TIP120

3.1.3 Résistor 10kΩ

On aura besoin de 3 résistances 10kΩ.



FIGURE 12 – Résistor 10KΩ

Paramètres de la résistance

1er anneau de couleur
Marron 1

2e anneau de couleur
Noir 0

3e anneau de couleur
Noir 0

Multiplicateur
Rouge $\times 100 \Omega$

Tolérance
Or $\pm 5\%$

Resistance value
10000 Ω

Sortie

Valeur de résistance:
10k Ohms 5%

[Rechercher dans le catalogue](#) [Effacer la sélection](#)

FIGURE 13 – Résistor 10K Code CouleurΩ

3.1.4 Battery 9V



FIGURE 14 – Battery 9V

3.1.5 BreadBoard

Pour tester le circuit avant de réaliser le prototype.

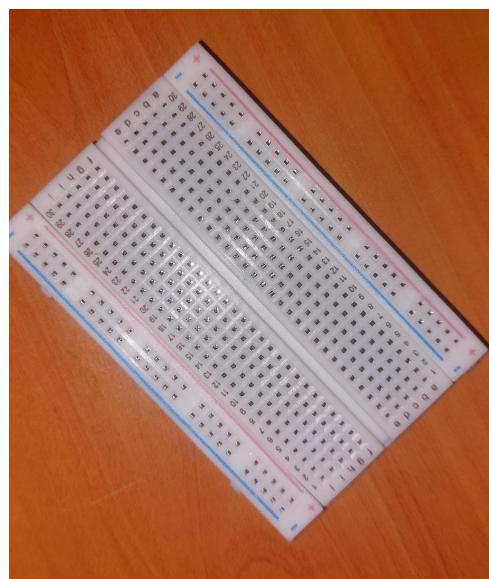


FIGURE 15 – BreadBoard

3.1.6 RGB LED Strip 12V



FIGURE 16 – RGB LED Strip 12V

3.1.7 fils



FIGURE 17 – fils

3.1.8 Carte de circuits imprimés

Pour servir de structure pour le prototype final.

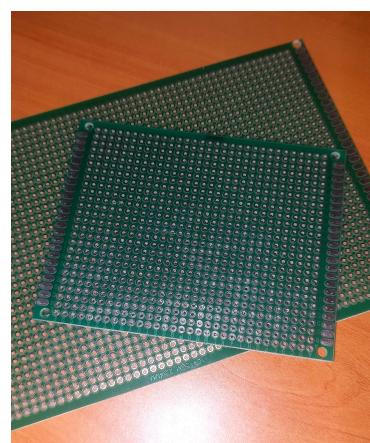


FIGURE 18 – Carte de circuits imprimés

3.2 Théorie de fonctionnement

Une bande LED RGB de cinq mètres fonctionnant sous 12 volts a été connectée à une carte ESP32. Étant donné que la carte Esp32 ne peut pas piloter directement la bande LED, nous avons utilisé un transistor NPN (TIP120) pour alimenter la bande LED à partir de l'Esp32. Les transistors sont connectés aux broches compatibles PWM de la carte de développement ESP32, chacune étant dédiée aux lignes rouge, verte et bleue de la bande LED.

Dans la partie programmation, nous avons créé un serveur web avec une page web statique qui se charge dans le navigateur lorsque nous saisissons l'adresse IP de l'ESP32. La page web contient des boutons permettant d'allumer/éteindre chaque couleur et d'autres boutons pour mélanger les couleurs rouge, verte et bleue afin de créer des combinaisons de couleurs variées.

3.3 Circuit

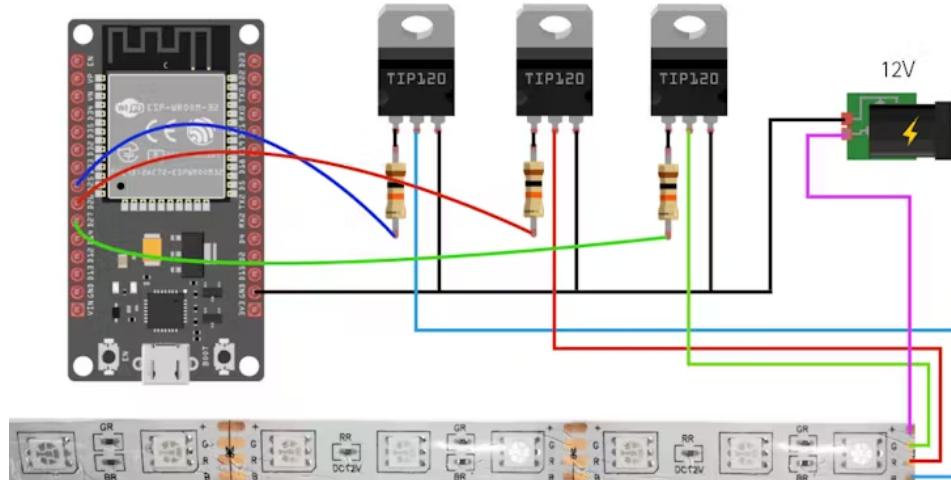


FIGURE 19 – Circuit du contrôleur

- Connectez les bornes R, G et B de la bande LED au collecteur des transistors, comme indiqué.
- À travers une résistance de $10\text{ k}\Omega$, connectez la base des transistors aux broches D25, D26 et D27 de l'ESP32.
- Tous les circuits de masse doivent être reliés ensemble, comme illustré : la masse de l'ESP32, la masse des transistors et la masse de l'alimentation 12V utilisée pour la bande LED RGB.
- Alimenter la carte Esp32 soit via le câble USB, soit via la borne VIN.

3.4 Code pour programmer l'ESP32

Voici le code source :

Listing 1 – Code pour le contrôle RGB avec ESP32

```
1 #include <WiFi.h>
2 #include <driver/ledc.h> // Include this for ESP32 PWM control
3
4 // Replace with your network credentials
5 const char* ssid = "Quarks";
6 const char* password = "houssam123";
7
8 // Set web server port number to 80
9 WiFiServer server(80);
10
11 // Variables to store RGB pin numbers and PWM channels
12 const int redPin = 26;
13 const int greenPin = 27;
14 const int bluePin = 25;
15
16 int redValue = 0;
17 int greenValue = 0;
18 int blueValue = 0;
19
20 // Function to generate HTML content
21 String generateHTML(int redValue, int greenValue, int blueValue) {
22     String html = "<!DOCTYPE html><html><head>
23         "<meta name=\"viewport\" content=\"width=device-width,
24             initial-scale=1\">
25         "<link rel=\"stylesheet\" href=\"https://cdnjs.
26             cloudflare.com/ajax/libs/materialize/1.0.0/css/
27                 materialize.min.css\">
28         "<style>body{font-family:Arial,sans-serif;text-
29             align:center;margin:20px;}"
30         ".color-preview{width:100px;height:100px;margin:20pxauto;border:1pxsolid#ccc;}"
31         ".button-group{margin:10px0;}</style></head><body>
32         "<h1>ESP32 RGB Controller</h1>
33         "<div class=\"color-preview\" style=\"background-color:rgb(" + String(redValue) + "," + String(greenValue) +
34             "," + String(blueValue) + ")></div>
35         "<h5>Adjust Colors</h5><div class=\"button-group\">
36             "<label>Red</label>
37             "<a href=\"/-r\" class=\"btn red\">-</a>
38             "<a href=\"/Red\" class=\"btn red lighten-1\">Toggle (" + String(redValue) + ")</a>
39             "<a href=\"/+r\" class=\"btn red\">+</a></div>
40             "<div class=\"button-group\">
41                 "<label>Green</label>
42                 "<a href=\"/-g\" class=\"btn green\">-</a>
43                 "<a href=\"/Green\" class=\"btn green lighten-1\">Toggle (" + String(greenValue) + ")</a>
44                 "<a href=\"/+g\" class=\"btn green\">+</a></div>
45                 "<div class=\"button-group\">
46                     "<label>Blue</label>
47                     "<a href=\"/-b\" class=\"btn blue\">-</a>"
```

```

44         "<a href=\"/Blue\" class=\"btn_blue_lighten-1\">" + String(blueValue) + "</a>" +
45         "<a href=\"/+b\" class=\"btn_blue\">" + "</a></div>" +
46         "</body></html>";
47     return html;
48 }

49

50 void setup() {
51   Serial.begin(115200);

52   // Connect to Wi-Fi network with SSID and password
53   Serial.print("Connecting to ");
54   Serial.println(ssid);
55   WiFi.begin(ssid, password);
56   while (WiFi.status() != WL_CONNECTED) {
57     delay(500);
58     Serial.print(".");
59   }
60   Serial.println("");
61   Serial.println("WiFi connected.");
62   Serial.println("IP address: ");
63   Serial.println(WiFi.localIP());
64   server.begin();

65   // Configure PWM channels
66   ledcSetup(0, 5000, 8); // Channel 0, 5 kHz, 8-bit resolution
67   ledcSetup(1, 5000, 8); // Channel 1, 5 kHz, 8-bit resolution
68   ledcSetup(2, 5000, 8); // Channel 2, 5 kHz, 8-bit resolution

69   // Attach PWM channels to pins
70   ledcAttachPin(redPin, 0);
71   ledcAttachPin(greenPin, 1);
72   ledcAttachPin(bluePin, 2);
73 }

74

75 void loop() {
76   WiFiClient client = server.available(); // Listen for incoming
77   clients

78   if (client) { // If a new client connects
79     Serial.println("New Client.");
80     String currentLine = ""; // Make a String to hold incoming data
81     from the client
82     String header = ""; // Variable to store the HTTP request
83     header

84     while (client.connected()) { // Loop while the client is connected
85       if (client.available()) { // If there's bytes to read from the
86         client
87         char c = client.read(); // Read a byte
88         header += c;
89         if (c == '\n') { // If the byte is a newline character
90           if (currentLine.length() == 0) {
91             // HTTP headers always start with a response code (e.g.,
92             // HTTP/1.1 200 OK)
93             // and a content-type so the client knows what's coming,
94             then a blank line
95           }
96         }
97       }
98     }
99   }
100 }
```

```

94     client.println("HTTP/1.1 200 OK");
95     client.println("Content-type:text/html");
96     client.println("Connection: close");
97     client.println();
98
99     // RGB control logic
100    if (header.indexOf("GET/-r") >= 0) redValue = max(0,
101        redValue - 1);
102    else if (header.indexOf("GET/+r") >= 0) redValue = min(255,
103        redValue + 1);
104    else if (header.indexOf("GET/+g") >= 0) greenValue = min
105        (255, greenValue + 1);
106    else if (header.indexOf("GET/-g") >= 0) greenValue = max(0,
107        greenValue - 1);
108    else if (header.indexOf("GET/+b") >= 0) blueValue = min
109        (255, blueValue + 1);
110    else if (header.indexOf("GET/-b") >= 0) blueValue = max(0,
111        blueValue - 1);
112    else if (header.indexOf("GET/Red") >= 0) redValue = (
113        redValue == 0) ? 255 : 0;
114    else if (header.indexOf("GET/Green") >= 0) greenValue = (
115        greenValue == 0) ? 255 : 0;
116    else if (header.indexOf("GET/Blue") >= 0) blueValue = (
117        blueValue == 0) ? 255 : 0;
118
119    // Update PWM values
120    ledcWrite(0, redValue);
121    ledcWrite(1, greenValue);
122    ledcWrite(2, blueValue);
123
124    // Send HTML response
125    client.println(generateHTML(redValue, greenValue, blueValue)
126        );
127    break;
128 } else {
129     currentLine = ""; // Clear the currentLine
130 }
131 } else if (c != '\r') { // If the character is not a carriage
132     return;
133     currentLine += c; // Add it to the currentLine
134 }
135 }
136
137 // Clear the header and stop the connection
138 header = "";
139 client.stop();
140 Serial.println("Client Disconnected.");
141 }
142 }
```

3.5 Montage en utilisant la "BreadBoard"



FIGURE 20 – Montage en utilisant la "BreadBoard"

3.6 La démonstration

On branche l'adaptateur USB de l'Esp32 pour l'alimenter.

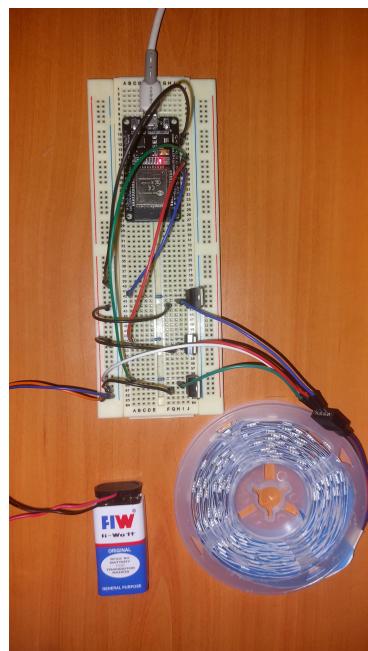
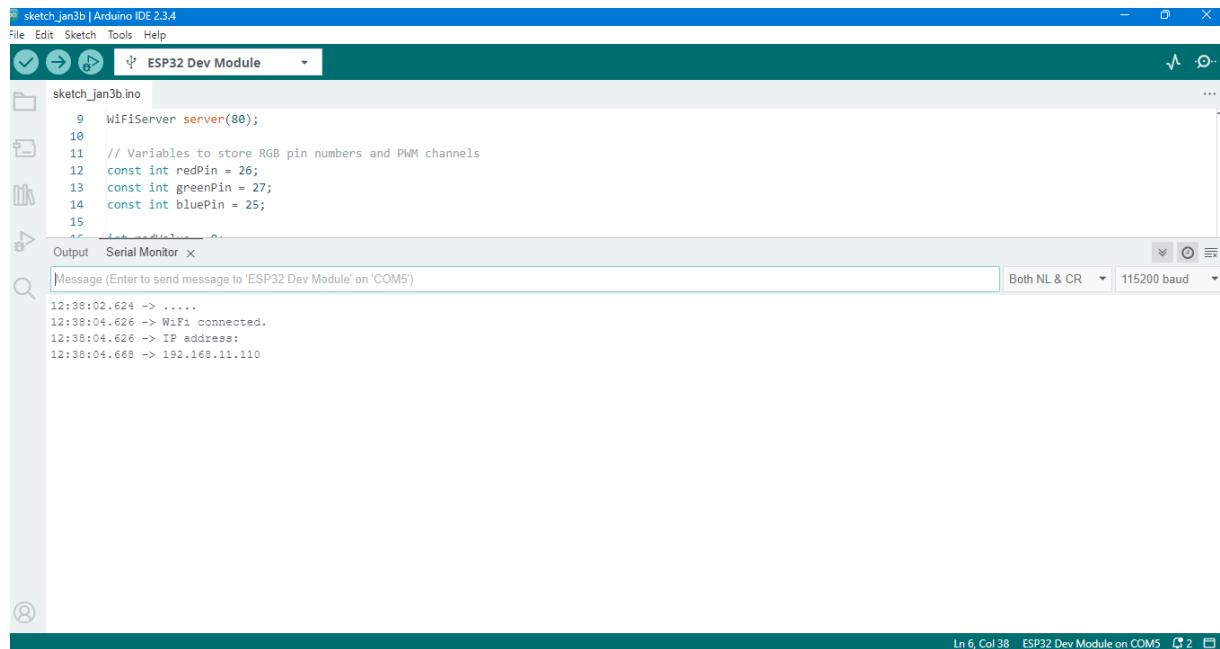


FIGURE 21 – Montage en utilisant la "BreadBoard"

Puis on ouvre le serial monitor dans ArduinoIDE.



```

sketch_jan3b | Arduino IDE 2.3.4
File Edit Sketch Tools Help
ESP32 Dev Module
sketch_jan3b.ino
9   Wifiserver server(80);
10
11 // Variables to store RGB pin numbers and PWM channels
12 const int redPin = 26;
13 const int greenPin = 27;
14 const int bluePin = 25;
15
Output Serial Monitor x
Message (Enter to send message to 'ESP32 Dev Module' on 'COM5')
12:38:02.624 -> .....
12:38:04.626 -> WiFi connected.
12:38:04.626 -> IP address:
12:38:04.668 -> 192.168.11.110
Ln 6, Col 38  ESP32 Dev Module on COM5  2  □

```

FIGURE 22 – Serial Monitor (ArduinoIDE)"

Et on se connecte sur l'adresse IP de l'esp32 (dans notre cas 192.168.11.110).

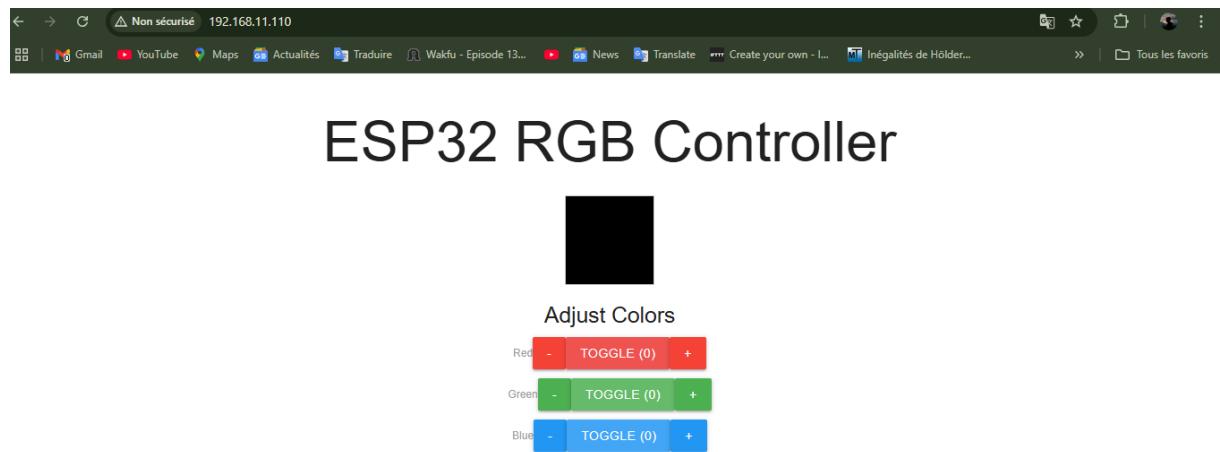


FIGURE 23 – Le site web pour contrôler l'Esp32

On clique par exemple sur le bouton rouge pour mettre les leds rouges sur le mode HIGH ou On (1).

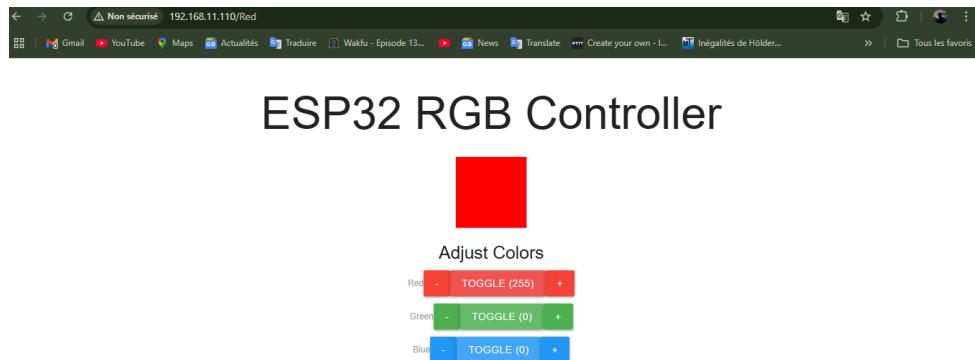


FIGURE 24 – LED rouge état ON : Site Web



FIGURE 25 – LED rouge état ON : Circuit

- On peut avoir la couleur vert en désactivant le bouton rouge et en activant le bouton vert.
- On peut avoir la couleur bleu en désactivant les boutons rouge , vert et en activant le bouton bleu.
- On peut faire des combinaisons de couleurs en activant plusieurs boutons en même temps.

La vidéo de la démonstration : Cliquez Ici

4 Réalisation et Prototype

4.1 Réalisation

On procède à la soudure du circuit sur la carte de circuits imprimés cité précédemment.

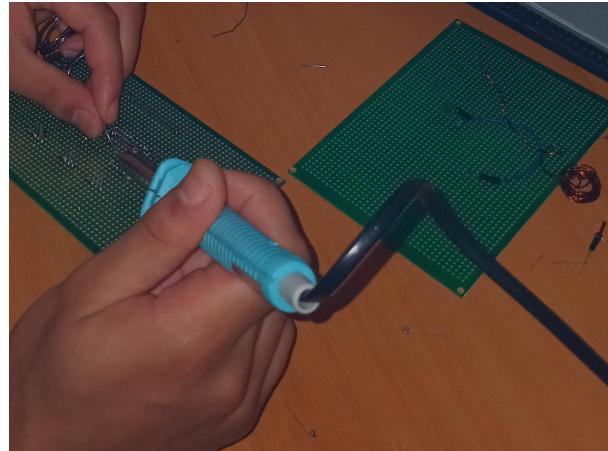


FIGURE 26 – Soudure

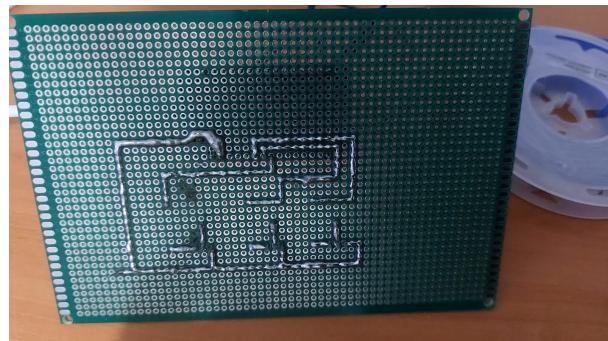


FIGURE 27 – Circuit soudé

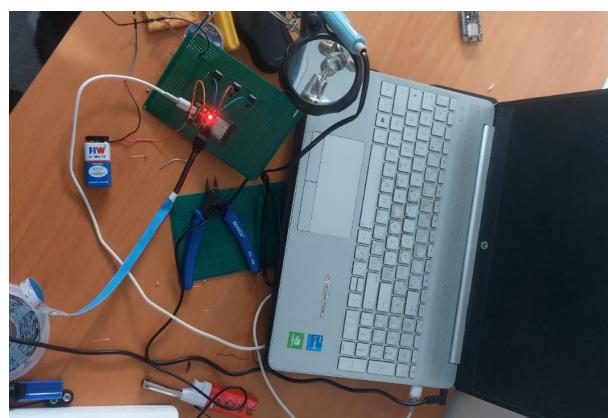


FIGURE 28 – Circuit global

4.2 Prototype final

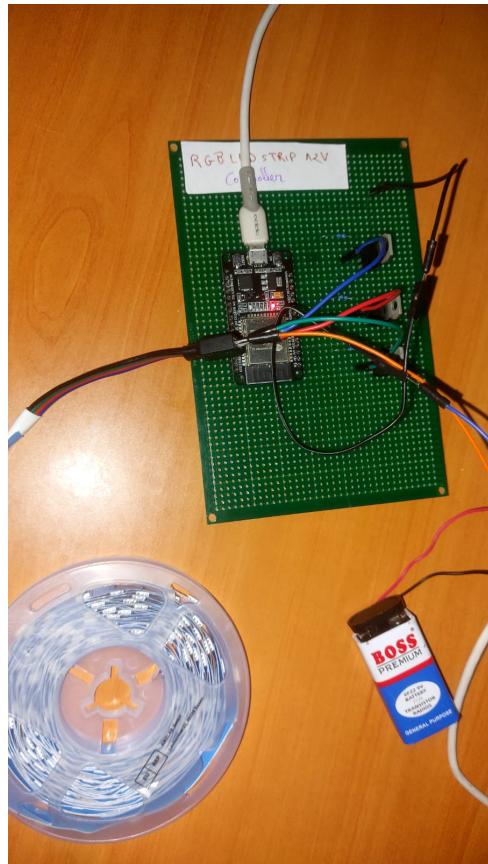


FIGURE 29 – Prototype final

[La vidéo du prototype final : Cliquez Ici](#)

5 Conclusion

Ce projet d'électronique analogique a permis de concevoir et de réaliser un contrôleur pour une bande LED RGB 12V, pilotée par une carte ESP32. Cette solution combine à la fois des composants analogiques, comme les transistors NPN (TIP120) pour la gestion de la puissance, et des outils numériques pour le contrôle via un serveur web intégré.

Grâce à cette réalisation, il est possible de commander les couleurs et les effets lumineux de la bande LED en temps réel à partir d'une interface web conviviale. Ce projet a non seulement permis de renforcer nos compétences en électronique analogique, mais aussi de mieux comprendre les interactions entre les systèmes de puissance et les systèmes numériques.

En guise de conclusion, ce projet illustre comment des concepts simples en électronique peuvent être intégrés pour concevoir des solutions pratiques et innovantes, tout en ouvrant la voie à des améliorations futures, comme l'ajout de modes d'éclairage programmables.

A Annexe :

Références

- [1] RGB controller, <https://www.hackster.io/najad/controlling-rgb-lights-from-esp32-web-server-f93620>.
- [2] ESP32, <https://www.upesy.com/blogs/tutorials/esp32-pinout-reference-gpio-pins-ultimate-guide?shpxid=12d4cad5-faef-4457-b875-931f675cb901>.
- [3] RGB LED STRIP, <https://www.instructables.com/Intro-to-LED-Strips/>.
- [4] Transistor TIP120, <https://www.onsemi.com/pdf/datasheet/tip120-d.pdf>.
- [5] Calculateur code couleur résistor, <https://www.digikey.com/en/resources/conversion-calculators/conversion-calculator-resistor-color-code>.
- [6] Guide couleur résistor, <https://neurophysics.ucsd.edu/courses/physics120/resistorcharts.pdf>.
- [7] How to Solder on Circuit Boards, <https://www.youtube.com/watch?v=Pjsoq-ZejRM>.