



# arbre de Noël circulaire 2023

célébration high-tech  
des fêtes de fin d'année

**Ton Giesberts (Elektor)**

La WS2812D-F8 est une LED RVB de 8 mm de haute technologie, programmable numériquement et adressable individuellement avec jusqu'à 255 dispositifs en guirlande. Dans ce projet de sapin de Noël circulaire en 3D, 36 de ces dispositifs peuvent être commandés de l'extérieur ou par un Arduino Nano ESP32 intégré. Les effets de lumière sont étonnantes – ne manquez pas d'avoir ce kit lumineux d'Elektor pour les fêtes de fin d'année !

Cet arbre de Noël se distingue par la forme de sa construction et l'utilisation de LED RVB numériques de 8 mm. Il ressemble plus à un arbre réel que la plupart des circuits imprimés (PCB) plats avec un contour en forme de sapin. Au lieu d'une simple conception en 2D, il s'agit d'une véritable création en 3D, basée sur les deux versions précédentes [1] et [2]. Le nouvel arbre est réalisé en séparant les cinq sections annulaires concentriques du PCB fourni avec le kit, et en les empilant avec des fils rigides pour espacer les cartes, donnant à ce projet l'aspect typique d'un arbre en forme de conifère.

Le projet [1] utilisait un générateur de nombres aléatoires simple avec deux circuits intégrés logiques standard et de petites LED CMS blanches. Le projet [2] reposait sur un microcontrôleur avec des LED connectées en matrice 6x6. Cette nouvelle version utilise 36 LED RVB numériques de 8 mm de type WS2812D-F8. Cela simplifie le circuit. Au lieu d'une matrice, les entrées et sorties des LED numériques sont connectées en série, chaque LED étant reliée à une alimentation en 5 V. La chaîne de LED peut être adressée comme une bande de LED NeoPixel. De nombreux programmes sur le web, dans de multiples langages, expliquent en détail comment gérer ces dispositifs.

Le connecteur K1 est relié au +5 V (4,3 V) interne, à la masse et à l'entrée de la première LED (par le biais du cavalier JP1). De cette manière, différents microprocesseurs, microcontrôleurs ou modules externes peuvent être utilisés pour commander les LED de notre arbre de Noël circulaire. Pour rendre l'arbre indépendant d'un circuit externe, on peut installer un module Arduino Nano ESP32 [3] sur le PCB de base.

## Schéma

Le circuit se compose essentiellement de 36 LED RVB numériques de type WS2812D-F8, dont les broches d'entrée ( $D_{in}$ ) et de sortie ( $D_{out}$ ) sont connectées en série, chacune d'entre elles étant alimentée par une alimentation de 4,3 V (figure 1). La tension réelle sur les LED est plus faible en raison de la chute causée par les diodes de protection D1 et D2. Les LED peuvent être commandées par un microprocesseur externe, un microcontrôleur, un module ou un module Arduino Nano ESP32 optionnel (MOD1).

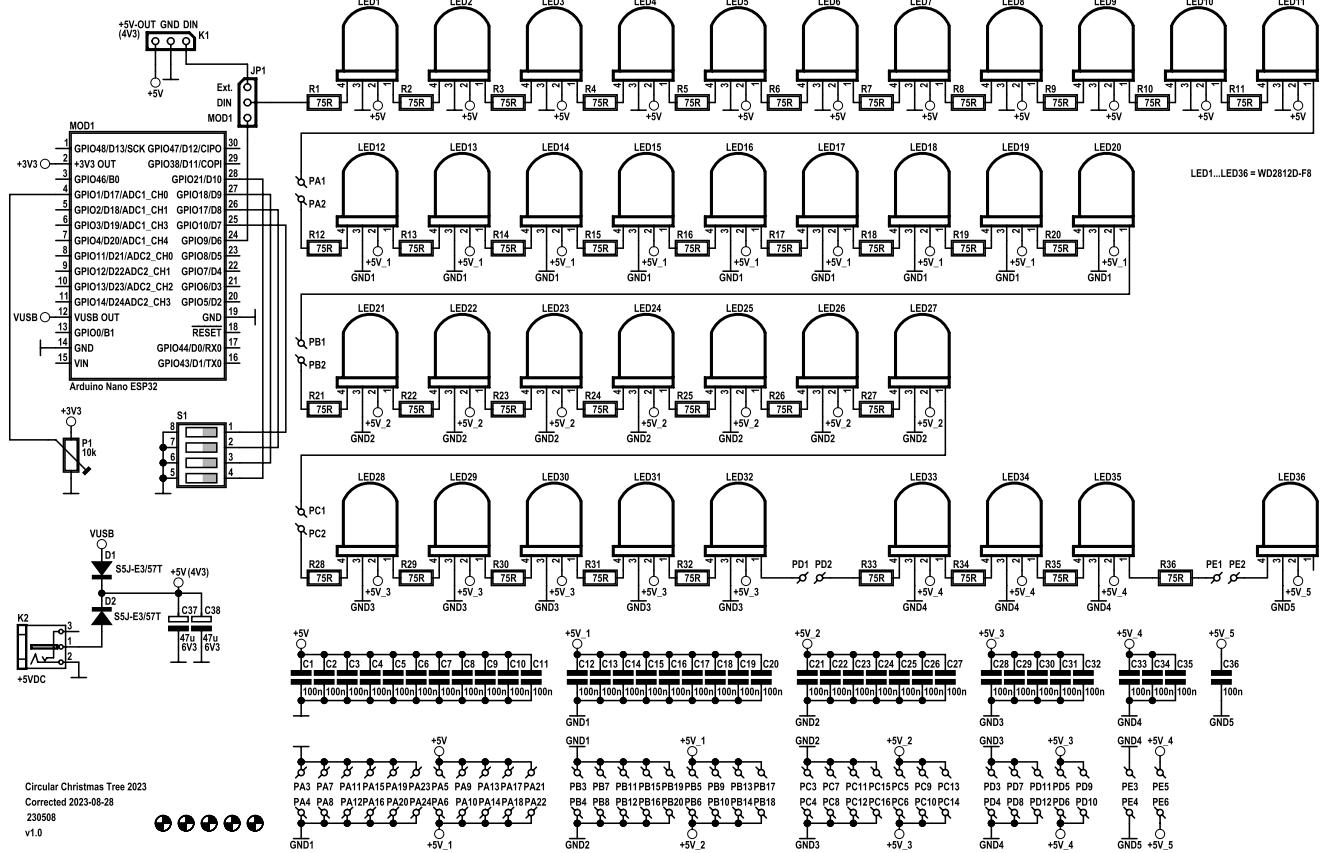


Figure 1. Schéma de l'arbre de Noël circulaire.

Le cavalier JP1, visible sur la **figure 2** près de la carte Arduino Nano, sélectionne le signal de commande de la LED1 (qui est la première des 36 LED en guirlande). L'utilisation de l'arbre est plus souple avec une carte Arduino Nano ESP32. On peut, par logiciel, modifier à distance les motifs d'éclairage de l'arbre en utilisant le Wi-Fi et/ou le Bluetooth LE intégrés. Toutes les entrées des LED ont une résistance de  $75\ \Omega$  en série (R1...R36, selon les exigences indiquées dans la fiche technique des WS2812D-F8 [4]). Chaque LED est découplée par un condensateur de 100 nF (C1...C36). De plus, les condensateurs au tantalum de 47  $\mu\text{F}$  C37 et C38 découpent l'alimentation des LED sur le PCB de base.

Tous ces composants sont des CMS et sont montés sur la face inférieure des PCB, donc invisibles. On peut alimenter les LED avec un adaptateur 5 VDC, de préférence d'au moins 1,5 A, via le connecteur d'alimentation CC K2, ou la broche VBUS de la carte Arduino Nano (si elle est présente) dans la zone du PCB de MOD1. Les diodes D1 et D2 (5 A, 50 VDC, S5J-E3/57T, boîtier CMS, taille SMC), également placées au-dessous du PCB, évitent les dommages si les deux alimentations sont connectées simultanément. On peut néanmoins connecter une alimentation à la fois au MOD1 et à K2, mais ce n'est pas nécessaire. Le module lui-même est alimenté par son connecteur USB-C, et le courant maximal de la broche VBUS est suffisant (en supposant que l'adaptateur secteur connecté soit assez puissant, bien sûr).

Les diodes ne sont volontairement pas de type Schottky. La chute de tension à travers les diodes est plus élevée et il faut s'assurer que les signaux logiques de 3,3 V sont bien dans la spécification des WS2812D-F8. Pour cette raison, si la chaîne de LED est commandée par un circuit externe, la meilleure option est toujours d'alimenter ce



Figure 2. Détail de MOD1, S1 et P1 montés sur le PCB de base.

circuit par la source d'alimentation interne de K1. La tension réelle sur K1 sera inférieure aux 5 V de K2 - autour de 4,3 V (ou même plus bas, avec un réglage élevé du courant/luminosité) - à cause de la chute de tension à travers D2.

Pour connecter les PCB, 39 morceaux de fils relient le +5 V et la masse d'un PCB à celui qui se trouve au-dessus (**figure 3**). Dans le schéma, il s'agit des connexions PA1-PA2 à PE5-PE6. Un fil supplémentaire sur chaque PCB relie la broche  $D_{out}$  de la dernière LED à la première LED du PCB au-dessus.



Figure 3. Un fil marqué et un autre dénudé. L'isolant restant doit avoir une longueur de 3 cm.

Si on utilise un Arduino Nano ESP32 pour commander les LED, on peut aussi installer sur le PCB le commutateur DIP S1 et le petit potentiomètre P1, tous deux visibles sur la **figure 2**. Le logiciel peut lire ces composants pour sélectionner différents motifs ou ajuster la luminosité. Ces composants sont inutiles sans le module.

## Alimentation électrique

L'alimentation des LED est assurée par un adaptateur CA vers +5 V CC avec un connecteur en bâillet, connecté à K2. Le kit disponible dans notre boutique ne contient que les composants pour le PCB, y compris le petit potentiomètre et les interrupteurs DIP. L'adaptateur secteur et le module optionnel Arduino Nano ESP32 ne sont pas inclus et doivent être achetés séparément.

Ce module dispose d'un connecteur USB-C qui alimente également les LED via sa broche VBUS. Comme mentionné ci-dessus, deux diodes (D1, D2) empêchent que les deux alimentations soient connectées directement l'une à l'autre. VBUS est connecté en interne au 5 V du connecteur USB-C par l'intermédiaire d'une diode Schottky de type PMEG6020AELR (Nexperia). Cette diode est prévue pour 2 A. Actuellement, le courant maximal de la broche VBUS n'est pas spécifié dans sa fiche technique [5]. Le courant des WS2812D-F8 est de 12 mA, selon une vue d'ensemble de cette famille sur le site web du fabricant.

Cependant, la fiche technique indique 15 mA dans ses spécifications, et de nombreux paramètres sont évalués pour  $I_F = 20$  mA. Dans notre prototype, le courant maximum au réglage maximum ( $3 \times 255$  en décimal par LED) était d'un peu moins de 11 mA par couleur – loin de 20 mA. Ces différences par rapport aux spécifications sont un peu déroutantes. Apparemment, une valeur plus réaliste pour le courant est d'environ 12 mA. En supposant que cette valeur soit correcte, le courant maximal à la luminosité maximale des LED serait de :

$$36 \text{ (DEL)} \times 3 \text{ (couleurs)} \times 12 \text{ mA} = 1,3 \text{ A.}$$

Dans notre prototype de sapin de Noël, le courant maximal total mesuré est de 1,156 A. La broche VBUS du MOD1 ne devrait pas avoir de problème avec ce courant. Mais, en général, nous vous conseillons vivement de ne jamais utiliser le courant nominal maximum d'une LED, car cela réduit considérablement sa durée de vie ! La luminosité entre, par exemple, un tiers et le courant nominal maximum n'est pas beaucoup plus faible. Pensez à en tenir compte lors de l'écriture du logiciel. Régler toutes les LED pour qu'elles produisent une lumière blanche, pour un éclairage très intense dans une pièce bien éclairée, nécessitera un courant global d'environ 200 mA. Cela sera plus que suffisant dans la plupart des cas et vous permettra d'utiliser n'importe quel port USB standard de type A pour alimenter le module.

## PCB

Comme prévu, le PCB de 136x136 mm est un panneau, le PCB de base contenant les cinq PCB de forme circulaire avec 24 ponts à casser. La carte entière figure dans l'encadré de la **liste des composants**, tandis que la **figure 4** montre toutes les parties séparées de cette carte, après avoir cassé les ponts. Sachez qu'il faut un peu forcer pour les casser. Un manuel de construction, disponible sur le site d'Elektor Labs [6], donne tous les détails sur la façon de construire l'arbre. Le cuivre des 39 morceaux de fil fait 0,8 mm d'épaisseur. Ces fils contribuent aussi à l'apparence de l'arbre, et c'est pourquoi nous avons choisi du fil isolé vert. Mais si vous trouvez qu'un fil nu étamé est plus joli, vous pouvez bien sûr enlever tout l'isolant.

Veillez à ce que tous les PCB soient fixés parallèles, avec un espace de 3 cm entre chaque carte. Le grand nombre de fils rend la construction assez rigide, comme on peut le voir sur la **figure 5**. Cette figure montre le prototype terminé sans le module supplémentaire. Pour que la chute de tension entre le bas et le haut soit minimale, tous les fils, à l'exception du fil de signal, sont connectés alternativement au +5 V et à la masse. Si vous ôtez tout l'isolant, prenez garde aux courts-circuits entre les fils dénudés !

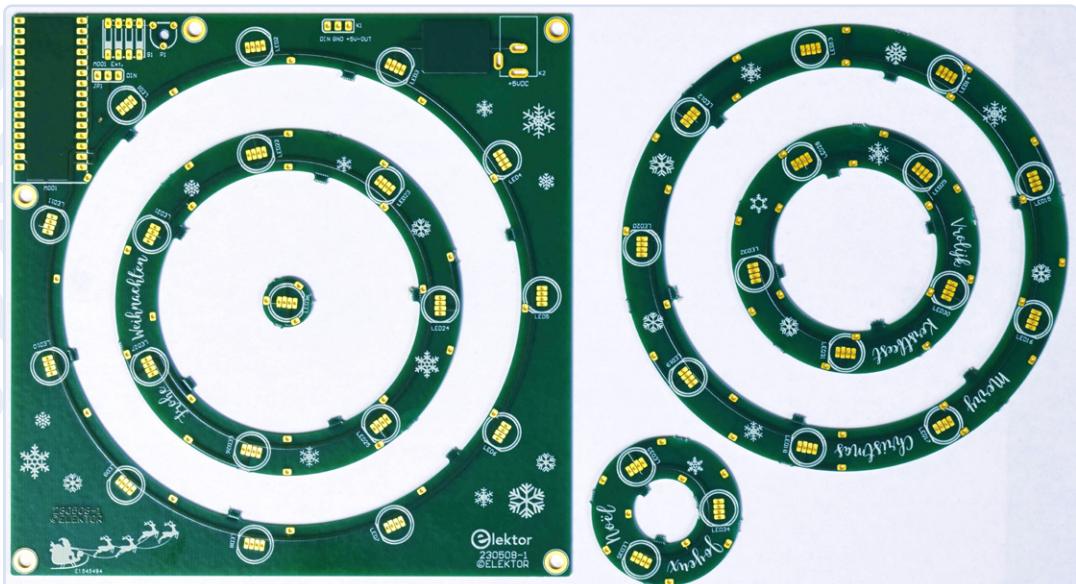
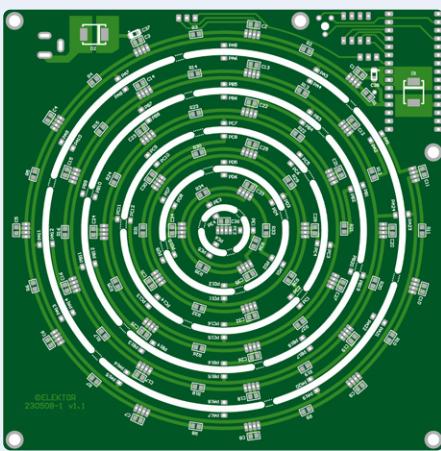


Figure 4. Tous les PCB séparés. Les ponts à casser qui reliaient les 6 PCB peuvent maintenant être retirés à l'aide d'une pince coupante.



## LISTE DES COMPOSANTS



### Résistances

R1...R36 = 75  $\Omega$ , 0W125, 5%, SMD 0805

P1 = potentiomètre circulaire 6 mm réglable par le dessus,  
10 k $\Omega$ , 0W1, 20%, (Piher PT6KV-103A2020)

### Condensateurs

C1...C36 = 100 nF, 50 V, 5%, X7R, SMD 0805

C37, C38 = 47 $\mu$ F, 6,3 V, 10%, tantale, taille de boîtier A (1206)

### Semi-conducteurs

D1, D2 = S5J-E3/57T, taille de boîtier SMD SMC

LED1-LED36 = WS2812D-F8, 8 mm, THT

MOD1 (optionnel) = Arduino Nano ESP32 avec barrettes

### Autres

K1, JP1 = barrettes, 3x1, vertical, pas de 2,54 mm

Cavalier de shunt pour JP1, pas de 2,54 mm

K2 = MJ-179PH (Multicomp Pro), connecteur d'alimentation CC, 4 A,  
diamètre des broches 1,95 mm

S1 = Interrupteur DIP, 4 voies

PA1...PE6 = 2 m de fil, 0,81 mm rigide, 0,52 mm<sup>2</sup> / 20AWG, isolé vert  
(Alpha Wire 3053/1 GR005)

H1...H5 = Entretoise en nylon, femelle-femelle, M3, 5 mm

H1...H5 = Vis en nylon, M3, 5 mm

### Divers

PCB 230508-1 (136 x 136 mm)

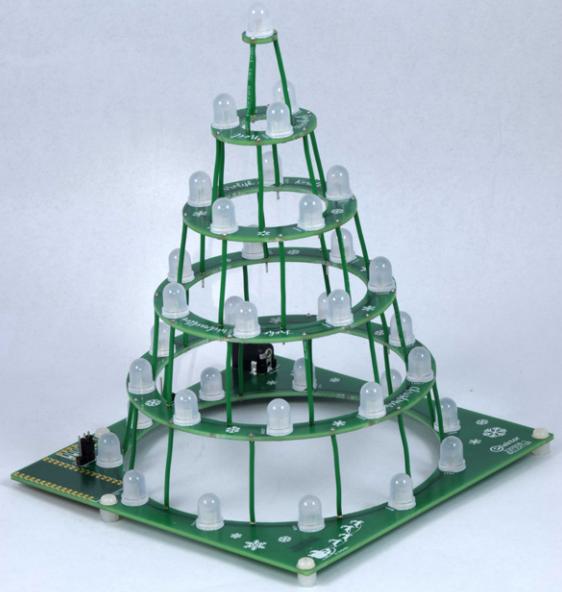


Figure 5. Le prototype terminé, sans le module Arduino optionnel.

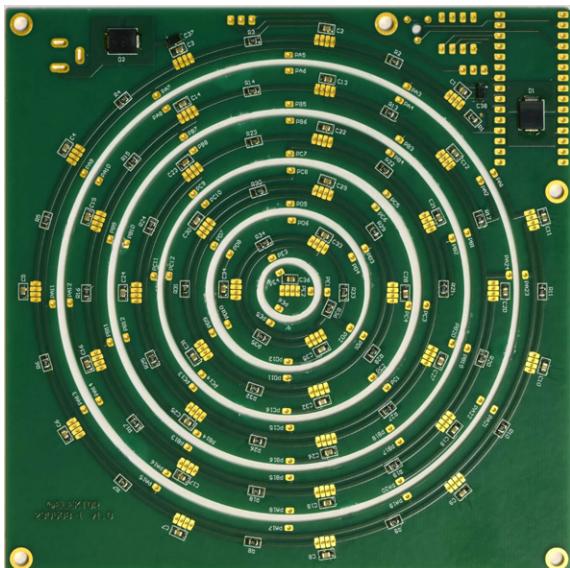


Figure 6. Tous les CMS sont soudés sur les faces arrière des PCB : R1...R36, C1...C38, D1 et D2.

Les pistes pour l'alimentation électrique font 1 mm de large. Tous les composants CMS sont montés sur la face inférieure des PCB, comme le montre la **figure 6**. Il faut séparer les PCB avant de commencer à souder. Il n'y a aucun autre composant sur le dessus des PCB circulaires, à l'exception des LED, dont les pattes doivent être coupées comme le montre la **figure 7**. Les composants à trous traversants JP1, K1, K2, P1, S1 et MOD1 sont placés sur le PCB carré de base. Comme ce projet utilise des composants CMS, il vaut mieux avoir une certaine expérience du soudage de ces composants.

Les résistances et condensateurs 0805 ne sont pas difficiles à souder à l'aide d'un fer à souder à panne fine et de la soudure fine. Une soudure très fine, d'un diamètre de 0,35 mm est recommandée, afin de ne pas utiliser trop d'étain pour les joints de soudure. L'arbre est placé sur cinq supports en nylon blanc de 5 mm fixés par cinq vis en nylon blanc. Ces vis sont moins visibles. Le module optionnel MOD1



Figure 7. Les fils de la LED RVB WS2812D-F8 sont coupés juste au-dessus de l'élargissement.

est placé dans un coin du PCB pour que la base soit aussi petite que possible. P1 et S1 sont placés à côté du module.

### Module optionnel Arduino Nano ESP32

Pour que cet arbre de Noël fonctionne indépendamment de tout circuit externe, on peut placer le module Arduino Nano ESP32 (MOD1) sur le PCB de base ainsi que le petit potentiomètre P1 et les commutateurs DIP S1. Placez le cavalier JP1 en position MOD1. Les deux composants supplémentaires peuvent être utilisés pour ajuster et/ou sélectionner la luminosité/vitesse et différents motifs/modes.

La programmation du module peut se faire avec la dernière version de l'EDI Arduino et le langage de programmation C. Si vous cherchez un projet MicroPython, cet arbre de Noël est également un bon choix. Un EDI dédié appelé Arduino Lab pour MicroPython [7] peut être téléchargé depuis GitHub. Un installateur MicroPython [8] est également nécessaire, et également disponible sur GitHub. Tout cela est bien documenté sur le site web d'Arduino et donne aux débutants de bonnes ressources pour le faire fonctionner [9].

Elektor a développé un logiciel simple téléchargeable à partir de la page web Elektor Labs de ce projet [6], où vous trouverez également des détails sur ses fonctionnalités.

Pendant le développement du logiciel, vous pouvez alimenter le projet à partir du connecteur USB d'un PC, mais un adaptateur CA-CC avec un connecteur UCB-C sera nécessaire pour alimenter l'arbre une fois terminé. Le courant maximum disponible dépend, bien sûr, du port spécifique. Grâce à un adaptateur USB-C vers USB-A ou à un câble adaptateur, l'Arduino Nano peut également être connecté à un port USB « ancien » si la luminosité est maintenue à un faible niveau. Le courant est alors limité à 500 mA, à retenir lors des tests du logiciel ! Vous pouvez également connecter une alimentation à K2 et régler la tension à un peu plus de 5 V afin d'alimenter les LED. La **figure 8** ne montre pas vraiment les couleurs vives des LED, mais donne une bonne impression de ce qui peut être obtenu avec différents réglages. 

VF : Denis Lafourcade — 230665-04

### À propos de l'auteur

Ton Giesberts a commencé à travailler chez Elektuur (aujourd'hui Elektor) après ses études, alors qu'on recherchait une personne ayant des affinités avec l'audio. Au fil des ans, il a surtout travaillé sur des projets audio. La conception analogique a toujours eu sa préférence. Bien entendu, les projets dans d'autres domaines de l'électronique font également partie de son travail. L'une des devises de Ton est la suivante : « Si vous voulez que ce soit mieux fait, faites-le vous-même ».

Par exemple, pour la conception d'un PCB destiné à un projet audio avec des taux de distorsion de l'ordre de 0,001%, une bonne disposition est cruciale !

### Questions ou commentaires ?

Contactez Elektor (redaction@elektor.fr).



### Produits

➤ **Kit sapin de Noël circulaire**  
[www.elektor.fr/20672](http://www.elektor.fr/20672)

➤ **Arduino Nano ESP32 avec connecteurs**  
[www.elektor.fr/20529](http://www.elektor.fr/20529)

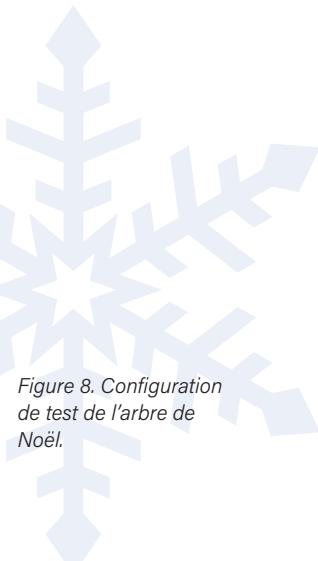


Figure 8. Configuration de test de l'arbre de Noël.

### LIENS

- [1] Arbre de Noël V1 : <https://elektormagazine.fr/labs/130478-xmas-tree-2014>
- [2] Arbre de Noël V2 : <https://elektormagazine.fr/labs/circular-christmas-tree-150453>
- [3] Documentation sur l'Arduino Nano ESP32 : <https://docs.arduino.cc/hardware/nano-esp32>
- [4] Fiche technique WS2812D-F8 [PDF] : [https://soldered.com/productdata/2021/03/Soldered\\_WS2812D-F8\\_datasheet.pdf](https://soldered.com/productdata/2021/03/Soldered_WS2812D-F8_datasheet.pdf)
- [5] Fiche technique Arduino Nano ESP32 [PDF] : <https://docs.arduino.cc/resources/datasheets/ABX00083-datasheet.pdf>
- [6] Ce projet sur Elektor Labs : <https://elektormagazine.fr/labs/circular-christmas-tree-2023-230508>
- [7] Arduino Lab pour Windows [Zip] : <https://tinyurl.com/arduinolab4win>
- [8] Arduino MicroPython Installer : <https://github.com/arduino/lab-micropython-installer/releases/tag/v1.2.1>
- [9] Cours MicroPython - MicroPython 101 par Arduino : <https://docs.arduino.cc/micropython-course/>