



# arbre de Noël 3D

circuit imprimé 3D avec un microcontrôleur 32 bits peu coûteux

André Araújo (Brésil)

Les nuits s'allongent et les vitrines étincelantes des centres commerciaux attirent le regard, tout en risquant de creuser votre budget. Pourtant, il est possible de recréer cette magie lumineuse chez soi avec des créations maison. En utilisant des composants électroniques abordables, comme un microcontrôleur à seulement 10 ¢, vous pouvez obtenir un éclat similaire sans vous ruiner. Que pensez-vous de cette approche innovante pour illuminer votre intérieur avant les fêtes de Noël ?

En octobre de l'année dernière, le projet brésilien Franzininho [1], une initiative dédiée au matériel informatique open source, a lancé un défi spécial pour Noël. L'objectif de ce concours était de concevoir des circuits imprimés basés sur un microcontrôleur abordables alimenté par une pile bouton, tout en privilégiant une faible consommation d'énergie. Un projet de base a été conçu et partagé sur le dépôt GitHub de Franzininho [2]. Les critères de base à respecter étaient les suivants :

- Microcontrôleur WCH CH32V003 RISC-V
- Pile CR2032
- Animations LED ou autres sorties graphiques
- En option : boutons, buzzers, etc.
- Circuit imprimé conçu en utilisant KiCad

Les participants étaient invités à s'inspirer du projet et à créer leurs propres versions,

idéalement en fabriquant leurs propres circuits imprimés. Mon idée consistait en un circuit imprimé conçu en forme de sapin de Noël, assemblé de manière transversale pour former une structure 3D. Ce projet peut faire office d'objet de décoration ou d'ornement à accrocher sur un sapin.

## Microcontrôleur

Avec les avancées dans le domaine des semi-conducteurs, les fabricants de microcontrôleurs introduisent désormais des modèles 32 bits, plus récentes et plus abordables, qui rivalisent avec et remplacent souvent les anciens microcontrôleurs 8 bits. Les puces basées sur le cœur ARM Cortex-M0+, telles que la série STM32C0 de STMicroelectronics, les séries LPC800 et MCX C de NXP, ou le Raspberry Pi RP2040, sont emblématiques de cette évolution. WCH, un fabricant chinois connu pour

ses émetteurs-récepteurs USB/UART WCH340 [3], a récemment introduit une nouvelle gamme de microcontrôleurs RISC-V, la série CH32V. Ces microcontrôleurs sont extrêmement abordables, certains modèles étant proposés à seulement 10 cents - ce qui leur a valu le nom de « 10-cent MCU ». Plusieurs membres de la communauté s'étaient déjà penchés sur ce nouveau microcontrôleur, faisant du défi de Noël une occasion de rassembler les passionnés autour d'un objectif commun : approfondir leur compréhension de ces puces et échanger leurs connaissances. Un élément crucial du défi était le matériel. Bien que la majorité des personnes impliquées dans la conception de systèmes embarqués se concentrent sur le logiciel, laissant la conception matérielle aux mains de quelques passionnés, l'accès à des logiciels de CAO performants, libres et gratuits, tels que KiCad [4] et EasyEDA [5], et à des services

de fabrication et d'assemblage de circuits imprimés abordables rend ce défi accessible. Il n'est pas nécessaire d'être un expert doté de compétences approfondies ou de moyens financiers conséquents pour y participer.

## Mon projet

Les cartes de mon projet comportent uniquement deux couches de signaux, ce requiert seulement des connaissances de base en matière de conception de circuits imprimés. Pour éviter les collisions de composants lors de l'assemblage final, j'ai utilisé un modèle papier pour vérifier les interférences et visualiser la disposition avant de procéder à la fabrication. Une méthode simple pour ce projet 3D consiste à imprimer les faces avant et arrière du circuit imprimé à l'échelle 1:1, les coller sur un morceau de carton fin, découper les contours et puis assembler les deux parties. Cela garantit l'exactitude des emplacements et du positionnement des composants. J'ai développé le logiciel en C en utilisant la couche d'abstraction matérielle (HAL) fournie par WCH, incluse dans MounRiver Studio [6]. Bien que la documentation relative à cette couche HAL soit peu fournie, l'archive *CH32V003EVT.ZIP* [7], disponible sur le site web de WCH contient des exemples pour les périphériques du microcontrôleur, qui sont suffisants pour comprendre les bases de la couche HAL et du microcontrôleur. La fiche technique du microcontrôleur [8] et le manuel de référence [9] sont également utiles.

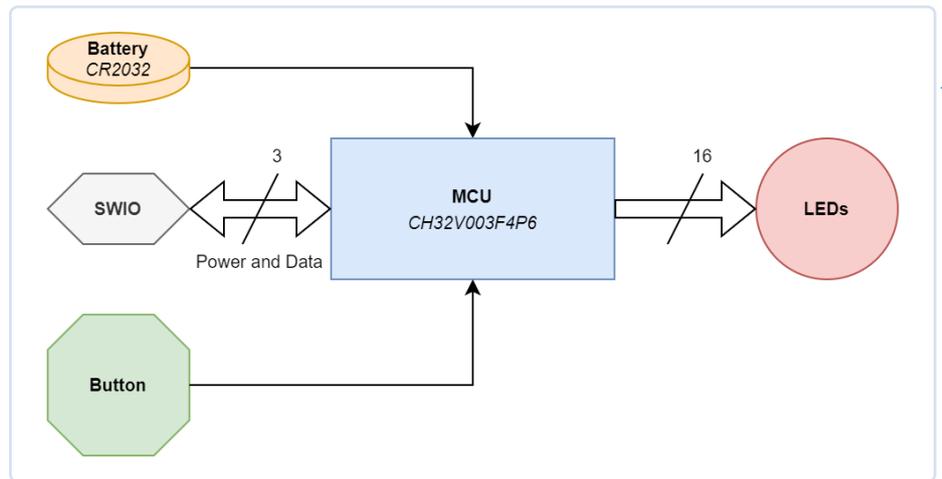


Figure 1. Schéma fonctionnel de l'arbre de Noël

Deux techniques logicielles sont importantes : le mode veille et le fonctionnement des LED avec un PWM logiciel. Ceci est crucial car seules certaines broches prennent en charge la génération PWM matérielle. Ces deux techniques contribuent à minimiser la consommation d'énergie.

## Schéma fonctionnel

Le matériel est divisé en plusieurs parties principales (voir **figure 1**) :

- **Microcontrôleur** : Le composant clé est le microcontrôleur CH32V003F4P6 [10] avec une fréquence d'horloge allant jusqu'à 48 MHz, 16 KB de mémoire flash, 2 KO de RAM, 18 lignes E/S, deux temporisateurs 16 bits, un temporisateur chien de garde, un ADC 10 bits à 8 canaux, un USART, un I<sup>2</sup>C et un SPI. L'alimentation est comprise entre 2,7 à 5,5 V. La puce est disponible en boîtier TSSOP20.

- **Pile** : Le PCB dispose d'un connecteur pour les piles CR2032 pour fournir 3 V pour l'alimentation.
- **Connexions** : Un connecteur à trois pôles permet la connexion de l'alimentation externe et fournit une communication bidirectionnelle pour la programmation et le débogage.
- **Interface utilisateur** : Un bouton utilisateur permet de modifier les animations ou de mettre l'appareil en veille.
- **Affichage** : 16 LED sont connectées au microcontrôleur, chacune à sa propre broche, permettant de nombreuses animations.

## Matériel

Le montage est simple : chaque LED est connectée à une broche GPIO individuelle via des résistances de limitation de courant. L'alimentation fournie par la pile est directement reliée au microcontrôleur, qui reste en

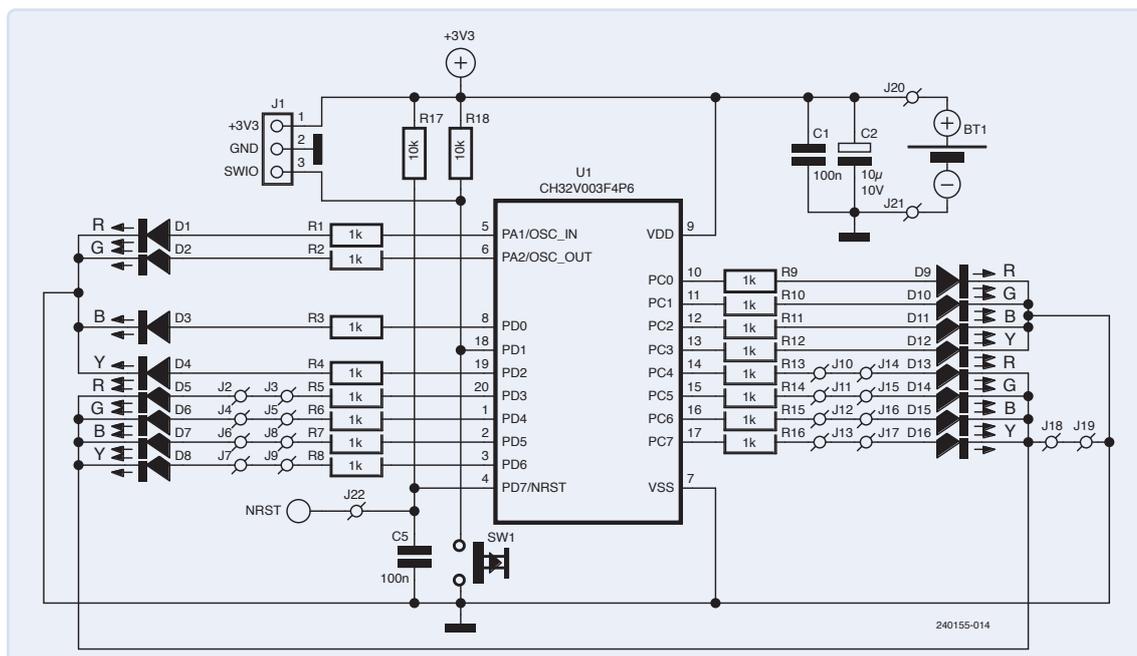


Figure 2. Le schéma relativement simple de l'arbre de Noël.

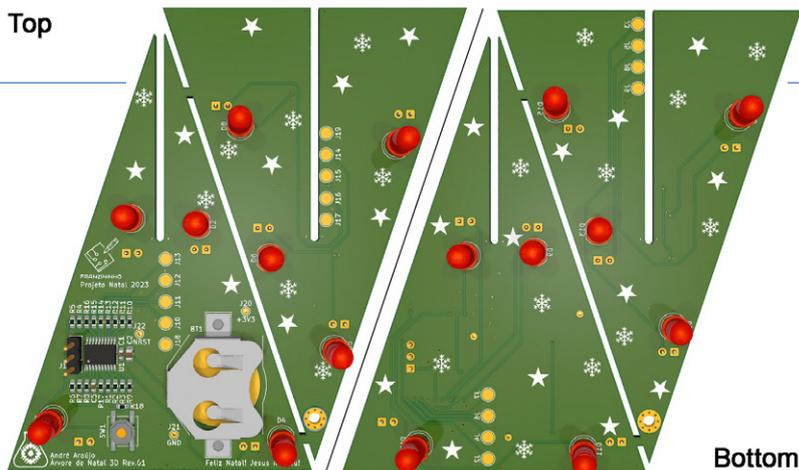


Figure 3. Rendu 3D des faces supérieure et inférieure du circuit imprimé peuplé.



Figure 4. WCH-Link, l'outil utilisé pour la programmation et le débogage. (Source : WCH [15])

mode veille la plupart du temps. Des paires de cavaliers sur le circuit imprimé facilitent l'assemblage des deux moitiés en 3D. Le schéma est illustré dans la **figure 2**. Il est important de noter que le bouton est connecté à la ligne SWIO (accessible via le connecteur J1), utilisée pour la programmation et le débogage. Ainsi, lorsque le bouton est pressé, le débogage devient inaccessible. Le circuit RC R17/C5 assure une réinitialisation lors de la mise sous tension. La broche de réinitialisation NRST est accessible via J22 sur le circuit imprimé.

La **figure 3** représente un rendu 3D des faces supérieure et inférieure peuplées du circuit imprimé conçu avec KiCad. Les deux moitiés sont reliées par deux petites languettes qu'il convient de couper pour les séparer. Les deux longues rainures dans chaque moitié permettent d'assembler les deux parties, formant ainsi la forme 3D du dispositif final. Des cavaliers de soudure, situés à proximité de ces rainures, doivent être soudés pour établir les connexions électriques nécessaires entre les deux moitiés.

## Logiciel

J'ai développé le micrologiciel avec MounRiver Studio [6], une version personnalisée de l'EDI Eclipse, qui prend en charge plusieurs microcontrôleurs de WCH et inclut la couche HAL pour accélérer le développement du logiciel. Vous pouvez télécharger les fichiers sources sur [14].

La **figure 4** montre le WCH-Link [11], un outil spécialisé pour la programmation et le

débogage du microcontrôleur. Une extrémité de cet outil se connecte à un PC via USB tandis que l'autre se termine par un connecteur destiné à être relié à un microcontrôleur. Il intègre également un convertisseur USB/UART, très utile pour le développement embarqué. Je n'ai pas utilisé cette fonctionnalité dans le cadre de ce projet.

Le programme offre quatre modes d'animation :

- **Entrelacé** : Deux couleurs sont allumées simultanément. À mesure qu'une couleur s'éteint, une autre s'allume dans un ordre séquentiel (par exemple, jaune et rouge, rouge et orange, orange et bleu, bleu et jaune, etc.).
- **Séquentiel** : Chaque couleur s'allume à tour de rôle, jusqu'à ce que toutes soient allumées. Elles s'éteignent ensuite dans le même ordre, jusqu'à ce que toutes les LED soient éteintes.

## Liste des composants

### Résistances

R1...R16 = 1 k, 1%, SMD 0603  
R17, R18 = 10 k, 1%, SMD 0603

### Condensateurs

C1, C5 = 100 n / 50 V, SMD 0603  
C2 = 10 μ / 25 V, SMD 0805

### Semi-conducteurs

D1, D5, D9, D13 = LED, rouge, 5 mm  
D2, D6, D10, D14 = LED, verte ou orange, 5 mm  
D3, D7, D11, D15 = LED, bleu, 5 mm  
D4, D8, D12, D16 = LED, jaune, 5 mm  
U1 = CH32003F4P6, TSSOP20

### Divers

BT1 = 3V CR2032  
SOCKET CR2032  
SW1 = TPS526, C&K (Littelfuse)  
Circuit imprimé

- **Clignotant** : Chaque couleur clignote individuellement, une à la fois.
- **Pulsation** : Chaque couleur s'allume et s'éteint séparément, et l'effet de fondu est réalisé par un PWM logiciel avec une largeur d'impulsion variable.

Chaque mode d'animation se répète un certain nombre de fois, puis passe au suivant. À la fin de la dernière animation, le microcontrôleur passe en mode veille. Le bouton peut être utilisé pour faire avancer le mode d'animation et mettre l'appareil en mode veille. En mode veille, le fait d'appuyer sur le bouton provoque une interruption qui réveille le microcontrôleur et reprend la première animation.

## Assemblage et tests

Après avoir finalisé la conception du circuit imprimé, j'ai commandé dix cartes auprès d'un fabricant chinois avec les composants

## Le projet Franzinho

Le projet Franzinho [1] est une initiative brésilienne centrée sur les systèmes embarqués, l'IdO et l'éducation des maker via des ressources matérielles et logicielles open-source. Il vise à promouvoir la connaissance, à partager l'expertise sur les technologies actuelles et à inspirer les individus à développer leurs propres projets. Actuellement, le projet a développé plusieurs cartes de développement, y compris des circuits basés sur l'ESP32-S2 (Franzinho WiFi et Franzinho WiFi LAB), STM32C0 (Franzinho C0), entre autres. Chaque design est fourni avec une documentation exhaustive et des exemples de code, facilitant l'accès tant pour les débutants que pour les développeurs plus expérimentés. Les cartes peuvent être programmées via des plateformes accessibles aux débutants telles qu'Arduino, MicroPython et CircuitPython, ou sur des systèmes plus complexes comme ESP-IDF, STM32CubeIDE, Zephyr et NuTTX.



Figure 5. Trois arbres de Noël prêts. L'un d'eux brille déjà.

CMS déjà assemblés - à l'exception du microcontrôleur, en rupture de stock. À réception, j'ai soudé le microcontrôleur et les composants traversants (LED et connecteur à trois pôles) et j'ai finalement assemblé les deux moitiés. La **figure 5** montre trois arbres de Noël entièrement fonctionnels.

Pour évaluer la consommation d'énergie de l'appareil final, j'ai utilisé un Nordic Power Profiler Kit II. Cet outil de test est capable d'alimenter l'appareil avec une tension réglable et également de mesurer le courant consommé par l'appareil. La figure 5 montre également la configuration du test. Les cartes ont été testées avec une alimentation de 3 V fournie par l'outil de test (sans pile durant les tests). L'énergie consommée est principalement due aux LED. La capture d'écran de la **figure 6** montre que l'arbre de Noël consomme en moyenne 9 mA, ce qui signifie qu'une pile CR2032 typique durerait environ 24 heures. Lorsque le microcontrôleur passe en mode veille, toutes les LED s'éteignent et le courant consommé chute à une moyenne d'environ 9 µA, rendant inutile un interrupteur d'alimentation puisqu'une pile CR2032 peut durer théoriquement jusqu'à 25 000 heures, soit environ 2,8 ans, en mode veille !

Il serait peut-être judicieux d'utiliser des LED orange au lieu des vertes pour un meilleur contraste avec le circuit imprimé vert.

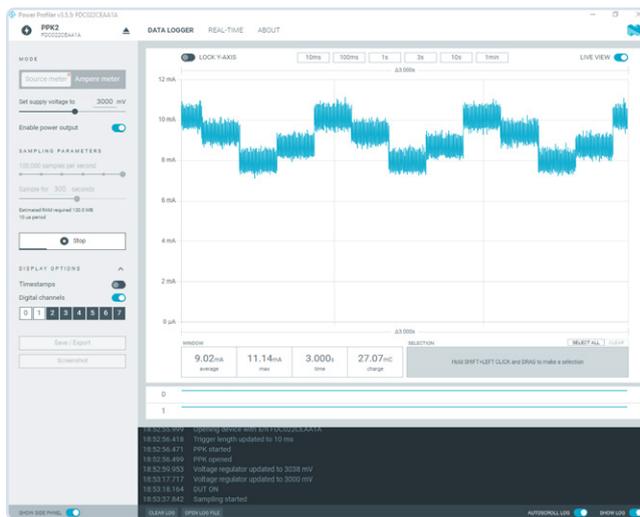


Figure 6. Capture d'écran du relevé de puissance fourni par Nordic Power Profiler Kit II.



### À propos de l'auteur

André Araújo a obtenu une maîtrise en ingénierie mécatronique à l'École polytechnique de l'Université de São Paulo et possède près de quinze ans d'expérience dans le développement d'équipements médicaux. Il occupe actuellement le poste de directeur de l'ingénierie chez J.G. Moriya. Par ailleurs, il est mentor pour le cours « Making Embedded Systems », dispensé par Elecia White sur la plateforme Classpert (<https://tinyurl.com/yc7us3ja>). André partage également des projets personnels sur son profil Instagram ([www.instagram.com/mechatronixlab](http://www.instagram.com/mechatronixlab)).

### Nouveau projet avec cette puce

Quatre modes d'animation ne sont qu'un début. Ce projet a permis à la communauté Franzininho, et maintenant à vous, de se familiariser avec ce nouveau microcontrôleur. Forts de cette première expérience collective avec la série CH32V, les membres de la communauté sont désormais impatients de se lancer dans un nouveau projet utilisant cette puce. Les préparatifs sont en cours et le lancement est prévu pour la fin de l'année !

240155-04



### Produits

- **ZD-8961-A Station de soudage à température contrôlée**  
[www.elektor.fr/20536](http://www.elektor.fr/20536)
- **Whadda WSPXL103 Poly Reindeer XL Soldering & Programming Kit (incl. Arduino Nano Every)**  
[www.elektor.fr/20325](http://www.elektor.fr/20325)

### Questions ou commentaires ?

Contactez l'auteur via LinkedIn ([www.linkedin.com/in/andremdaraujo](http://www.linkedin.com/in/andremdaraujo)) ou contactez Elektor ([redaction@elektor.fr](mailto:redaction@elektor.fr)).

### LIENS

- [1] Le projet Franzininho : <https://franzininho.com.br>
- [2] Page de Franzininho sur GitHub : <https://github.com/Franzininho/Projeto-Natal-2023>
- [3] Émetteurs-récepteurs WCH340 USB/UART : <https://www.wch-ic.com/products/CH340.html>
- [4] KiCad : <https://www.kicad.org>
- [5] EasyEDA : <https://www.easyeda.com>
- [6] MounRiver Studio : <http://www.mounriver.com>
- [7] Archive de CH32V003EVT : <https://tinyurl.com/49sfjzmr>
- [8] Fiche technique du CH32V003 : <https://tinyurl.com/7yzy5jab>
- [9] CH32V003 reference manual: <https://tinyurl.com/bde8tvmk>
- [10] Microcontrôleur CH32V003F4P6 : <https://www.wch-ic.com/products/CH32V003.html>
- [11] WCH-Link : <https://www.wch-ic.com/products/WCH-Link.html>
- [12] Nordic Power Profiler Kit : <https://tinyurl.com/3urhseh5>
- [13] Fichiers du projet sur GitHub : <https://tinyurl.com/54fdyuaa>
- [14] Ce projet sur Elektor Labs : <https://www.elektormagazine.fr/labs/3d-christmas-tree-pcb>
- [15] WCH-Link : <https://www.wch-ic.com/products/WCH-Link.html>

