

# régulateur de vitesse pour ventilateur ou aérateur avec modes manuel et thermostat

Bruno Clerc (France)

Il fait chaud dehors (ou à l'intérieur) et vous rêvez d'une brise bien fraîche ? Avec ce contrôleur de ventilateur, vous pouvez contrôler manuellement le flux d'air d'un ventilateur ou laisser l'appareil s'en charger.

L'été 2022 a été particulièrement chaud et, comme beaucoup d'entre vous, j'étais à la recherche d'une brise rafraîchissante. Je n'ai pas l'air conditionné (ni dans la maison, ni dans la voiture) mais j'avais deux ventilateurs de voiture/caravane de 12 V (**figure 1**), j'en ai branché un sur une alimentation électrique. Toutefois, comme ces ventilateurs tournent à plein régime dès qu'ils sont allumés, ils sont bruyants et soufflent un peu trop fort à mon goût. Ça m'a motivé pour réaliser ce contrôleur de vitesse de ventilateur.

L'article « vent de fraîcheur avec ATtiny » [1] a été une grande source d'inspiration, car il m'a aidé à gérer correctement la sortie MLI (en anglais PWM, *Pulse Width Modulation*) pour contrôler un ventilateur. Mes propres essais MLI ont fait « chanter » le ventilateur. En utilisant le Timer0 en mode MLI rapide comme décrit dans l'article, le ventilateur est devenu plus silencieux.

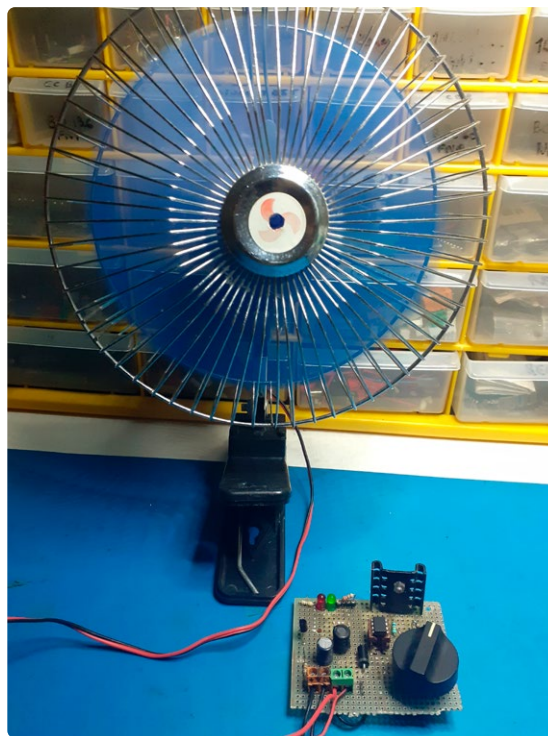


Figure 1. Le ventilateur qui a inspiré ce projet, contrôlé par un prototype sur carte perforée.

Le logiciel accompagnant l'article n'était pas écrit pour Arduino, je l'ai donc adapté en un croquis compilable avec l'EDI Arduino. Mon application n'avait pas besoin d'un capteur de température, je l'ai quand même gardé en option pour conserver l'esprit de l'article original. J'ai remplacé le pilote du capteur de température DS18B20 inclus dans le programme original par la bibliothèque OneWire pour Arduino.

## Circuit

Le schéma du contrôleur de ventilateur, illustré à la **figure 2**, est en partie basé sur l'article mentionné ci-dessus [1]. Mon premier prototype utilisait un MOSFET pour piloter le ventilateur, mais il perturbait les valeurs renvoyées par le capteur DS18B20, ce qui rendait l'ensemble instable. Peut-être les leçons tirées de [2] auraient-elles pu m'aider dans ce cas ? J'ai préféré utiliser un transistor de puissance à la place du MOSFET, et le circuit imprimé comporte donc un emplacement TO-218 pour Q1, ce qui m'a permis de recycler un transistor NPN récupéré sur un vieux équipement. Le courant de sortie maximum est de 2 A (limité par la self L1), ce qui peut nécessiter de monter le transistor sur un radiateur.

La résistance R5, qui se trouve entre l'alimentation 12 V et l'entrée de U2, un régulateur 5 V dans un boîtier TO-92, introduit une chute de tension qui réduit l'échauffement du régulateur. Le circuit consomme environ 30 mA dans sa version complète, R5 doit donc être capable de dissiper 117 mW.

Le capteur DS18B20 est relié à J3. La LED D3 signale s'il est présent ou non.

La diode Schottky D4 pourrait être d'un autre type. Les LED D2 et D3 sont optionnelles, de même que le bouton de réinitialisation SW1.

Un interrupteur (actif bas) peut être installé pour prendre le pas sur le potentiomètre de consigne. Le port PB4 du microcontrôleur est configuré en entrée avec une résistance de rappel. Notez qu'en cas d'utilisation d'un tel interrupteur, il faut supprimer la résistance R3 et la LED D3.

Pour cet interrupteur, j'ai utilisé un module à touche tactile TTP223 (J8). Ce module possède deux cavaliers de configuration à souder ; court-circuitez le cavalier A et laissez le cavalier B ouvert. Au repos, la sortie du module est à l'état haut. En le touchant, la sortie passe à l'état bas jusqu'à ce qu'on enlève le doigt.

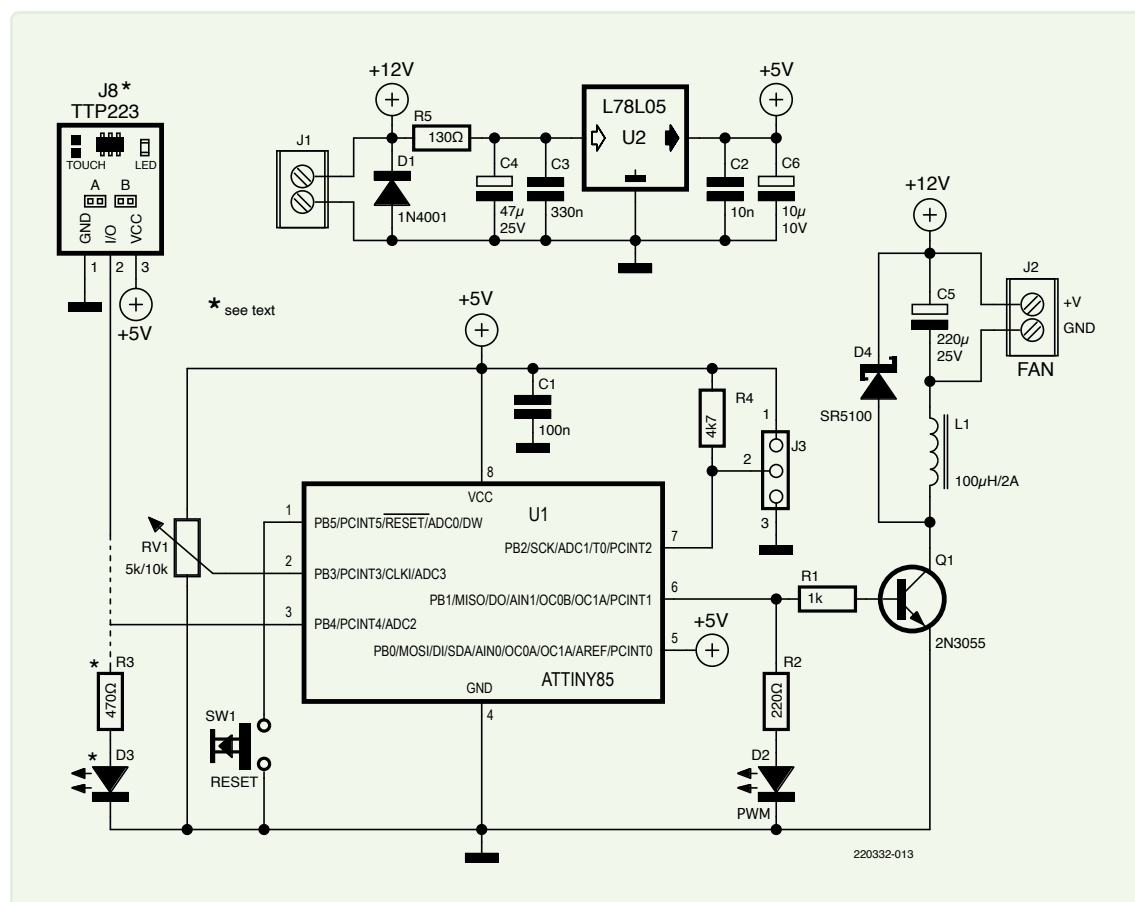
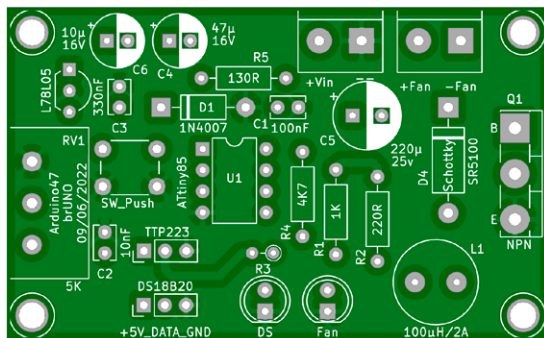


Figure 2. Schéma du contrôleur de la vitesse. Pour utiliser ce circuit comme thermostat, connectez un DS18B20 à J3. Montez également soit la touche TTP223 sur J8, soit R3/D3, mais pas les deux.

Figure 3. Un modèle de carte de circuit imprimé (PCB) est disponible à l'adresse [3].



## Logiciel

Le programme est configuré au moyen de quelques directives `#define` en tête pour :

- le choix entre l'interrupteur TTP223 ou la LED d'état du DS18B20.
- les seuils MLI `PWM_MIN` et `PWM_MAX`, ainsi que `PWM_PULSE_UP`, le coup d'accélérateur au départ pour garantir un bon démarrage du ventilateur.
- la plage de température si un capteur DS18B20 est présent, ainsi qu'une valeur de température maximale si le ventilateur doit toujours fonctionner à plein régime. Les valeurs de température sont spécifiées sous forme de nombres entiers à trois chiffres, par exemple 224 pour 22,4 °C. Dans cette version du logiciel, les valeurs de température négatives ne sont pas gérées, mais on peut l'ajouter, car le capteur fonctionne parfaitement en dessous de zéro.

Au démarrage, le croquis vérifie la présence ou non d'un capteur DS18B20 afin de déterminer le mode de fonctionnement : A ou B. En mode A (capteur absent), l'appareil est un simple contrôleur de vitesse manuel avec un potentiomètre de consigne. La valeur de la vitesse (0 à 100%) est obtenue par la fonction `map()`, qui convertit la valeur de la tension analogique du potentiomètre dans la plage 0 à 1023 à une valeur MLI de `PWM_MIN` à `PWM_MAX`. La valeur de `PWM_MIN` dépend du ventilateur et est la plus grande valeur qui ne le fait pas tourner. J'ai utilisé 30 ; j'ai réglé `PWM_MAX` sur 255.

En mode B, (capteur présent) le potentiomètre contrôle la température cible souhaitée. L'appareil fonctionne alors comme un thermostat. Le ventilateur se met en marche lorsque la température ambiante dépasse la valeur fixée par le potentiomètre. La plage de température est définie dans le programme par une valeur minimale et une valeur maximale. La fonction `map()` associe d'abord la valeur du potentiomètre à une température cible dans la plage de température. Ensuite, cette valeur est convertie en une valeur dans la plage MLI, comme dans le mode A.

Le logiciel utilise le chien de garde comme temporisation pour la lecture de la température ou la réactivation des touches.

Enfin, quelques notes sur la préparation de l'ATtiny85 pour ce projet :

- Programmez un chargeur de démarrage compatible Arduino sur l'ATtiny85.
- Le microcontrôleur doit fonctionner avec son oscilateur interne de 16 MHz.
- Installez le paquet ATtiny Boards de David Mellis pour programmer l'ATtiny85 avec l'IDE Arduino.

Tous les fichiers de conception, y compris le circuit imprimé (**figure 3**), peuvent être téléchargés à partir de [3].

VF : Helmut Müller — 220332-04

## Des questions, des commentaires ?

Contactez Elektor ([redaction@elektor.fr](mailto:redaction@elektor.fr)).

## À propos de l'auteur

Bruno Clerc découvre l'électronique vers l'âge de 12 ans, grâce à son frère aîné. Curieux de tout et assoiffé de connaissances, il décide de faire des études d'électronique à Bordeaux. Il a travaillé dans les systèmes basse tension tertiaires, puis dans l'aéronautique et dans divers autres emplois. À l'arrivée des microcontrôleurs, ne connaissant pas la programmation, il se concentre sur la maintenance de matériel hi-fi ancien. Tout change lorsqu'il y a quelques années, son frère lui offre un Arduino UNO. Bruno s'est trouvé une nouvelle passion et est devenu « Arduino47 ». Aujourd'hui, il remercie toute la communauté Arduino qui l'a aidé à progresser dans son apprentissage.



## Produits

- **W. A. Smith, Explore ATtiny Microcontrollers using C and Assembly Language**  
<https://elektor.fr/20007>
- **Fan SHIM – Refroidissement actif pour Raspberry Pi 4**  
<https://elektor.fr/19039>

## LIENS

- [1] Stephan Laage-Witt, « vent de fraîcheur avec ATtiny » Elektor 6/2016 : <https://www.elektormagazine.fr/magazine/elektor-201606/29036>
- [2] Stuart Cording, "Why Do MOSFETs Need Drivers?" Elektormagazine.com : <https://elektormagazine.com/articles/why-do-mosfets-need-drivers>
- [3] Ce projet sur Elektor Labs : <https://elektormagazine.fr/labs/variateur-for-ventilateur-or-fan>



# MagPi, le magazine officiel du Raspberry Pi

FAITES UN ESSAI MAINTENANT ET RECEVEZ UN  
**RASPBERRY PI PICO GRATUIT**  
POUR VOS PROPRES CIRCUITS D'ÉTÉ

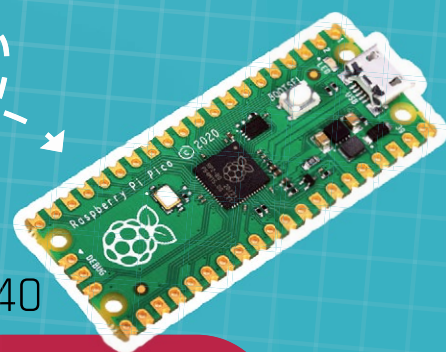


## Abonnement d'essai

- ✓ 2 x édition MagPi imprimée
- ✓ 4 mois d'accès digital à MagPi
- ✓ L'ensemble des archives françaises de MagPi
- ✓ Résiliation automatique



**La prime à l'essai :**  
Le Raspberry Pi Pico RP2040



**OBTENEZ CETTE OFFRE EXCLUSIVE À**  
[magpi.fr/circuits](https://magpi.fr/circuits)

avec le code : **CIRCUITS23**

\* Valable uniquement pour les nouveaux membres. Jusqu'à épuisement du stock.