

Écran tactile rond pour Raspberry Pi

L'HyperPixel 2.1 Round de Pimoroni



Clemens Valens
(Elektor)

Optimisé pour les Raspberry Pi Zero et Zero 2 W, cet écran tactile capacitif rond de 2,1 pouces (53 mm) de diamètre est idéal pour créer des appareils élégants destinés, par exemple, aux applications de mesures et de domotique.

L'HyperPixel 2.1 Round de Pimoroni [1] est un écran tactile capacitif IPS (In-Plane Switching) rond de 2,1 pouces avec DPI (Display Pixel Interface) à haute vitesse. Comme ses frères HyperPixel 4 carrés et rectangulaires, le 2.1 Round est destiné au Raspberry Pi. En fait, sa taille est optimisée pour les Raspberry Pi Zero et Zero 2 W mais, comme il possède le connecteur HAT (Hardware Attached on Top) standard à 40 broches, il peut être branché sur n'importe lequel équipé d'un tel connecteur, à condition de faire attention à l'aspect mécanique des choses.

Spécifications

La résolution de l'écran est de 480 par 480 pixels, mais comme il est rond, il faut bien sûr soustraire les coins. Il a une profondeur de couleur de 18 bits, soit 262 144 couleurs, et supporte jusqu'à 60 fps (images par seconde). La zone de visualisation a un diamètre de 2,1 pouces (53,3 mm) et un angle de vision de 175°. Son diamètre total est de 72 mm pour une hauteur de 11 mm. Avec un Pi Zero attaché, la hauteur totale (ou la profondeur, selon votre préférence) est de 17 mm.



Figure 1. L'HyperPixel 2.1 Round montrant le bureau du Raspberry Pi Buster.

Q : combien de pixels sont perdus à cause des coins arrondis ? *

Comme l'écran utilise presque toutes les broches du connecteur HAT, vous ne pouvez pas ajouter d'autres cartes d'extension. Cependant, l'écran fournit un port I²C alternatif pour y connecter d'autres circuits. La consommation d'un système HyperPixel + Raspberry Pi Zero 2 W est d'environ 300 mA en mode de fonctionnement normal. Lors du démarrage, j'ai observé des pics de plus de 500 mA, donc pour être sûr, vous auriez besoin d'une alimentation de 5 W, 5 V.

Pilote et Bibliothèques

Pour utiliser l'HyperPixel 2.1 Round sur un Raspberry Pi, vous devez d'abord installer un pilote. Des instructions détaillées sur la façon de le faire sont données sur le site Web de Pimoroni et dans leur coin GitHub [2]. Les pilotes sont uniquement pour Raspberry Pi OS Buster, mais le support pour Bullseye est en cours de développement. Même si Bullseye n'est pas encore supporté, j'ai fait un essai. Mon programme de démonstration (voir ci-dessous) a fonctionné de la même manière, sauf que le menu du bureau s'est superposé en haut de l'écran (**figure 1**). J'ai branché l'HyperPixel 2.1 Round sur un Raspberry Pi Zero 2 W exécutant Buster et j'ai activé SSH pour me faciliter la vie par la suite. Après avoir installé le pilote et redémarré, l'écran a montré une version minuscule du bureau, et j'ai pu ouvrir une fenêtre de terminal. Maintenant, le port HDMI ne fonctionne plus et l'HyperPixel est devenu le seul écran. Comme il est plutôt minuscule, j'ai fait la plupart du travail par SSH sur mon ordinateur portable Windows. Pour utiliser l'écran dans vos propres applications, vous avez besoin de la bibliothèque Python3 [3] de Pimoroni. Elle comprend quelques exemples, mais ils n'ont pas bien fonctionné pour moi. Après quelques recherches, j'ai découvert que la mise à niveau de la bibliothèque pygame vers la dernière version (2.1.0 au moment où je l'ai fait) a résolu les problèmes que j'avais. Après cela, toutes les démos ont bien fonctionné. Elles sont tactiles, donc touchez l'écran pour changer les couleurs.

Notez que le tactile nécessite un pilote pour fonctionner comme une souris sur le bureau. Malheureusement, un tel pilote ne semble pas encore exister, mais vous pouvez utiliser l'exemple *daemon* inclut dans la bibliothèque *uinput-touch.py* à la place. Cela fonctionne très bien.

Ma première application

Une fois que j'ai eu l'écran en marche, il était temps de voir si je pouvais en faire quelque chose. Mon idée était de l'utiliser pour mon compteur

* R : $480 \times 480 \times (1 - 0.25 \pi) = 49,444$ (c'est-à-dire presque 21.5%)



Figure 2. L'horloge compteur d'abonnés YouTube permet de faire glisser sa valeur.



Figure 3. L'HyperPixel 2.1 Round a le même diamètre que l'intérieur d'une boîte de Pringles.

d'abonnés YouTube [4] afin de remplacer l'affichage à 7 segments par quelque chose de plus sympa. La démo de l'horloge semblait être un bon point de départ. Tout ce que j'avais à faire était d'ajouter la partie requête de l'API YouTube pour obtenir le nombre d'abonnés et ensuite imprimer le nombre à l'écran.

Tout d'abord, j'ai installé `httplib2` qui est nécessaire pour le fragment du compteur d'abonnés YouTube. L'exécution du programme a révélé que j'avais également besoin de `libsd12-ttf` pour imprimer du texte à l'écran. Gardez à l'esprit que votre application doit appeler `pygame.init` si vous voulez travailler avec du texte et des polices et que vous devez en charger une. C'est plutôt long sur un Raspberry Pi Zero, donc je l'ai placé au début du programme. En utilisant la police, vous rendez le texte dans une image que vous copiez ou `blit` dans le tampon de l'écran pour le rendre visible (figure 2). Comme l'écran est tactile, j'ai ajouté une fonction permettant de positionner le compteur d'abonnés n'importe où sur l'écran en le faisant simplement glisser. Comme l'horloge, sa couleur dépend de l'endroit où vous touchez l'écran. Pour le faire ressortir, il n'utilisera pas la même couleur que l'horloge, sauf si vous touchez le centre pour que tout soit blanc.

Mon code peut être trouvé à ClemensAtElektor sur GitHub [5].

Remarques additionnelles

Un mot sur la rotation de l'écran. Je voulais le faire pivoter de manière à ce que les connecteurs USB et HDMI du Raspberry Pi Zero pointent vers le haut. Cela correspond à 180° ou *inverted*. Le pilote Pimoroni fournit des utilitaires pour faire cela, mais ils n'ont pas fonctionné pour moi. Ce qui a fonctionné, cependant, était simplement d'ajouter la ligne `display_lcd_rotate=2` au fichier `/boot/config.txt` et de redémarrer le système.

Notez que pour certaines raisons, le centre du tampon d'écran peut ne pas être exactement le centre de l'écran, il peut être décalé de plusieurs pixels dans la direction verticale. Vous pouvez corriger cela en ajoutant un décalage dont le signe dépend de la rotation de l'écran. Vous pouvez voir dans mon code comment j'ai géré cela.

Port I²C alternatif

J'ai fait un essai rapide avec le port I²C alternatif de l'écran. Selon le site web de Pimoroni, ce serait le port I²C 3, mais je l'ai trouvé comme port 11. Vous pouvez savoir ce que vous avez en lançant la commande `i2cdetect -l` ou en cherchant dans le dossier `/dev` les fichiers qui commencent par `i2c`.

J'y ai connecté un module accéléromètre Mabee MPU6050 [6] pour voir s'il serait trouvé. L'outil `i2cdetect` a montré un périphérique à l'adresse 68 hexadécimale, qui est effectivement l'adresse imprimée sur le module, donc cela semblait fonctionner. J'ai également pu lire le capteur après avoir installé la bibliothèque `mpu6050-raspberrypi`.

Il rentre exactement dans une boîte de Pringles !

J'ai remarqué que le diamètre de l'écran HyperPixel 2.1 Round est presque le même que celui d'une boîte de Pringles, il rentre exactement dedans. J'ai donc décidé de construire mon compteur d'abonnés YouTube dans une boîte de Pringles vide.

J'ai découpé le haut et le bas d'une canette et j'ai fixé l'ensemble écran-Raspberry-Pi à la partie inférieure avec des supports de sorte que l'écran vienne au même niveau que l'intérieur de l'anneau sur la partie supérieure. Maintenant, l'écran est protégé lorsque le couvercle est en place et le tactile continue de fonctionner (figure 3).

Résumé

Pour résumer, je pense que l'HyperPixel 2.1 Round est un complément très intéressant pour un Raspberry Pi Zero 2 W. Il fonctionne également sur un Zero. La qualité de l'image est très bonne, et le toucher fonctionne bien dans vos propres applications. Ce serait un ajout assez cool à un système de domotique.

Officiellement, il n'y a de support que pour Buster, mais je l'ai essayé sur Bullseye. Mon programme a fonctionné de la même manière, sauf que le menu du bureau s'est superposé en haut de l'écran. ◀

220358-04



PRODUITS

➤ HyperPixel 2.1 Round (SKU 19870)
www.elektor.fr/19870

LIENS

- [1] Site web de Pimoroni:
<https://shop.pimoroni.com/products/hyperpixel-round>
- [2] Coin GitHub de Pimoroni:
<https://github.com/pimoroni/hyperpixel2r>
- [3] Bibliothèque Python3:
<https://github.com/pimoroni/hyperpixel2r-python>
- [4] YouTube subscriber counter:
<https://youtu.be/PuCCMZFNqQE>
- [5] ClemensAtElektor sur GitHub:
<https://github.com/ClemensAtElektor/HyperPixel2r>
- [6] Module accéléromètre Mabee MPU6050:
www.makerfabs.com/mabee-6aixs-imu-mpu6050.html