

carte CLUE d'Adafruit

Une solution intelligente pour les projets IoT

Tam Hanna (Slovaquie)

La carte BBC micro:bit a connu un grand succès, bien au-delà du marché éducatif auquel elle était destinée. La carte CLUE d'Adafruit vient d'apparaître sur le marché, avec un écran à part entière et plus de mémoire. Dotée du Bluetooth LE et d'une multitude de capteurs intégrés, elle convient particulièrement bien aux petits projets IoT.

Après le succès des cartes Arduino et Raspberry Pi, il est devenu évident qu'il y a de l'argent à faire avec les différents types d'ordinateurs éducatifs. En 2016, la BBC est entrée dans la course avec micro:bit, un ordinateur monocarte doté d'un SoC Bluetooth de Nordic Semiconductor au lieu d'un processeur à part entière sous Linux.

Depuis lors, des entreprises ont vécu d'incroyables histoires, telles

que Spy oz en Slovaquie qui a lancé des sociétés qui ne s'occupent que de la distribution de l'écosystème micro:bit [1].

L'adage selon lequel « à peine est-on servi qu'on regarde ce qu'il y a dans l'assiette du voisin » s'applique également au domaine de l'informatique embarquée. Les progrès continus dans le domaine des SoC Bluetooth ont laissé les 16 MHz et les 16 Ko de SRAM de micro:bit paraître un peu dépassés. De plus, son afficheur à 5 × 5 LED

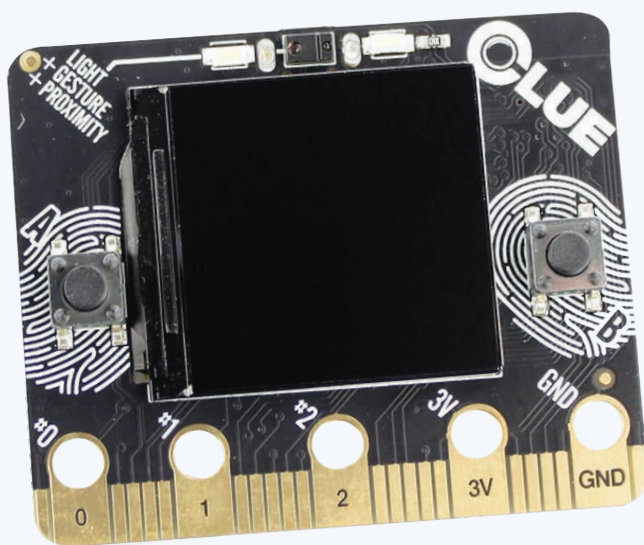


Figure 1. L'attaquant vu de face...

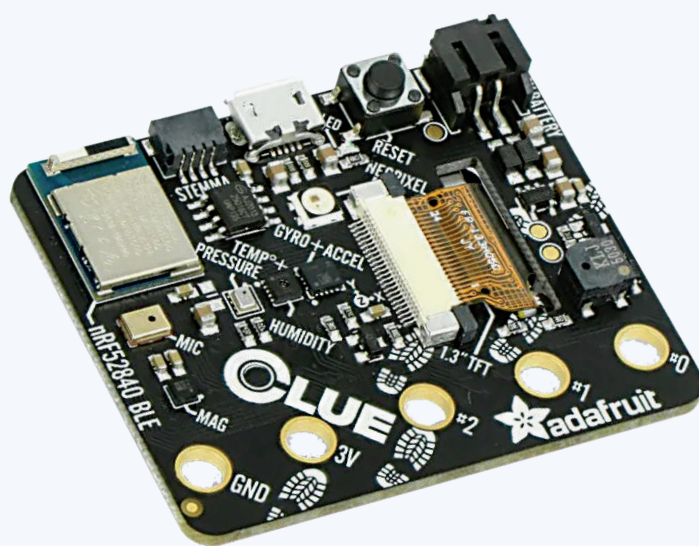


Figure 2. ... et par l'arrière.

CARACTÉRISTIQUES

- Processeur Nordic nRF52840 Bluetooth LE processor : 1 Mo de flash, 256 Ko de RAM, cœur à 64 MHz Cortex M4
- Écran TFT IPS de 1,3», 240×240, couleur pour texte et graphiques
- Alimentation : toute source de 3,6 V (régulateur interne et diodes de protection)
- Deux boutons pour l'utilisateur et un bouton de RàZ
- Capteur de mouvement, accéléromètre/gyroscope et magnétomètre
- Capteurs de proximité, lumière, couleur et mouvement
- Capteur de son PDM, microphone
- Capteur d'humidité SHT
- Capteur BMP280 pour température et pression barométrique/altitude
- LED RVB NeoPixel de signalisation
- Mémoire flash interne de 2 Mo pour enregistrement de données, images, fonts ou code CircuitPython
- Buzzer/haut-parleur pour l'émission de sons et de bips
- Deux LED blanches brillantes à l'avant pour l'éclairage/la détection des couleurs
- Connecteur Qwiic/STEMMA QT pour ajouter des capteurs, pilotes de moteurs ou des écrans par I²C. Possibilité de connecter des capteurs GROVE I²C avec un câble d'adaptation.
- Programmable avec l'EDI Arduino ou le langage CircuitPython.

ne convient pas pour représenter autre chose que les plus simples des graphiques.

Adafruit attaque avec CLUE

Avec le lancement du nRF52840 de Nordic Semiconductor, un SoC Bluetooth monocœur dont le processeur ARM atteint 64 MHz et est doté de 256 Ko de RAM, il y avait de quoi proposer un nouveau plat.

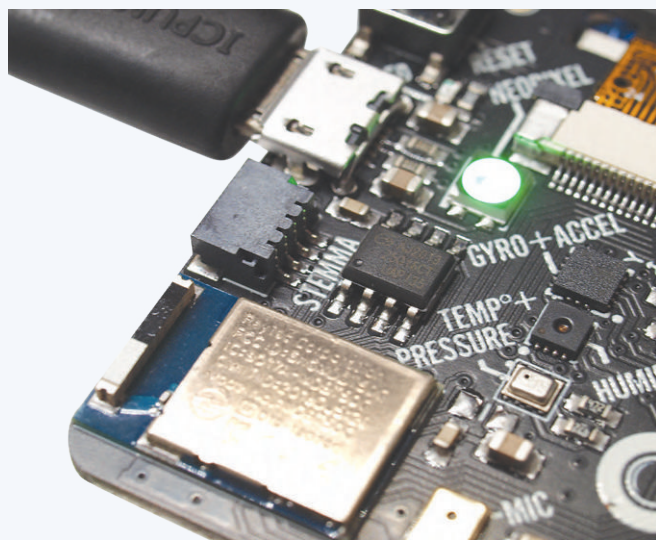


Figure 3. Le port STEMMA permet à la carte CLUE d'Adafruit d'être connectée à des cartes d'extension.

Les **figures 1 et 2** montrent le résultat : la carte CLUE d'Adafruit qui semble très similaire à la carte micro:bit de BBC. À part le SoC, qui n'est pas immédiatement visible sur ces photos, c'est l'écran beaucoup plus grand situé à l'avant qui se distingue vraiment. Les LED sont remplacées par un écran couleur de 240×240 pixels, basé sur la technologie classique IPS LCD plutôt que sur la technologie organique.

Un autre détail agréable de la carte est le connecteur placé à l'arrière, cf. **figure 3**. Elle dispose d'un bus I²C au format interne d'Adafruit sur lequel d'autres capteurs peuvent être facilement connectés. Il existe également un adaptateur pour le format Grove de Seeed, qui propose divers capteurs à prix raisonnable.

Notez que la carte CLUE ne semble être que partiellement compatible avec la micro:bit. Bien que le connecteur le long du bord inférieur soit d'une construction identique, l'utilisation d'un afficheur différent signifie qu'à première vue, la plupart des boîtiers disponibles pour le BBC micro:bit ne conviendront pas pour la carte CLUE d'Adafruit.

L'auteur a testé cette hypothèse avec un boîtier ThingiVerse de domw disponible sur [2]. La partie avant ne convenait pas, car l'écran de la CLUE était beaucoup plus grand que la matrice de LED de l'original de BBC. Compte tenu de cela, l'auteur a été particulièrement surpris par le fait que la face arrière du boîtier s'est correctement montée, malgré les connecteurs supplémentaires. Cependant, lorsqu'on l'examine de plus près, cela est probablement dû au fait que la conception du boîtier était relativement généreuse. Si le boîtier avait été conçu avec un ajustement plus serré, il aurait été probablement inutilisable.

Une question de programmation

Comme la micro:bit est un système éducatif, le développement ne ressemble pas à l'expérience que l'on peut avoir en utilisant les environnements de développement embarqués classiques tels que

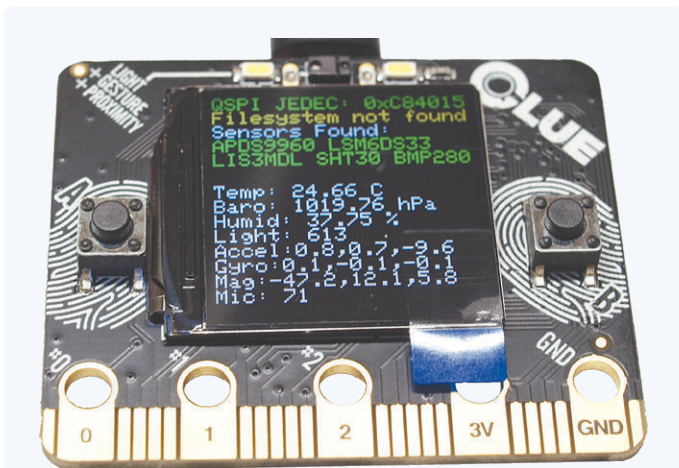


Figure 4. Cette appli pré-installée montre les informations renvoyées par les capteurs.

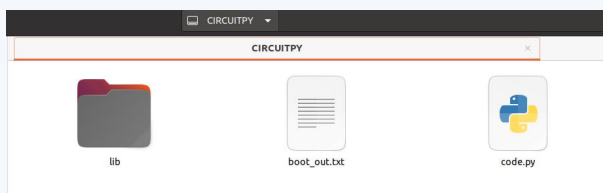


Figure 5. Le « runtime » Python dévoile également un lecteur virtuel.

ARM Keil. Cela peut être gênant pour les puristes de l'embarqué, mais en pratique il est indispensable, car de nombreuses universités n'ont pas assez de personnel compétent pour déboguer les programmes C++ (croyez l'auteur : les étudiants créent des erreurs de programmation vraiment complexes).

Au lieu de cela, on s'appuie généralement sur un quatuor composé de l'EDI Arduino, CircuitPython, MakeCode et Scratch. Cependant, pour la CLUE, seulement deux de ces environnements sont actuellement disponibles. MakeCode est en cours d'adaptation, sans date de disponibilité précise, et il n'y a aucune information pour Scratch. Un chargeur d'amorçage série pour déployer le code est inclus, un peu comme pour la carte Raspberry Pi Pico.

Pour une première petite expérience, faisons fonctionner CircuitPython. Si vous connectez une carte neuve à un ordinateur, via le connecteur micro-USB à l'arrière, l'écran affiche une page d'état (fig. 4) qui fournit des informations sur l'état fonctionnel.

En appuyant deux fois sur le bouton *reset* situé au dos de la carte, le tampon de trame du pilote de l'écran se fige. Le poste de travail connecté (l'auteur utilise Linux) voit alors une nouvelle clé USB où le code compilé peut être chargé.

Fait intéressant, la carte CLUE d'Adafruit est toujours visible par

l'ordinateur. Si elle n'est pas en mode *bootloader*, *dmesg* la détecte comme suit :

```
tamhan@TAMHAN18:~$ dmesg
...
[28292.202193] usb 1-2.7: Manufacturer: Adafruit LLC
[28292.202195] usb 1-2.7: SerialNumber: 7687A137B6FDB874
[28292.204040] cdc_acm 1-2.7:1.0: ttyACM0: USB ACM
device
```

Après une double pression sur le bouton *reset*, une clé USB apparaît, comme ici :

```
tamhan@TAMHAN18:~$ dmesg
...
[28371.624193] sd 10:0:0:0: Attached scsi generic sg6
type 0
```

Important : ce lecteur ne reste pas activé pour toujours. S'il demeure inutilisé pendant plus de 30 s environ, le micrologiciel se remet à tourner normalement.

Recherche de fichier

En visitant l'URL https://circuitpython.org/board/clue_nrf52840_express/, nous pouvons faire notre première étape et télécharger le fichier *adafruit-circuitpython-clue_nrf52840_express-en_US-6.1.0.uf2*. Il contient le *runtime* qui doit être placé sur la clé USB.

Notez que vous trouverez également un fichier nommé *CURRENT.UF2* sur le lecteur. Celui-ci vous permet de télécharger le micrologiciel qui se trouve actuellement dans la mémoire du système cible.

Étrangement, le *runtime* n'est pas fourni pas avec une bibliothèque complète spécifique à tous les capteurs disponibles. Nous devons plutôt aller dans l'URL <https://circuitpython.org/libraries> pour télécharger l'archive *adafruit-circuitpython-bundle-6.x-mpy-20210329.zip* puis l'extraire dans un dossier approprié du système de fichiers.

À ce stade, vous devriez jeter un autre coup d'œil à l'écran de la carte CLUE, car le *runtime* délivre en sortie et en permanence le contenu de la console. Un détail agréable est que le dispositif sur le PC (cf. fig. 5) présente la mémoire interne de l'environnement de travail Python.

Il est important de déplacer les dossiers suivants de l'archive vers le dossier *Libs* du dispositif :

```
adafruit_apds9960
adafruit_bus_device
adafruit_display_shapes
adafruit_display_text
adafruit_lsm6ds
adafruit_register
```

Comme si cela ne suffisait pas, Adafruit s'attend également à ce que vous rassembliez les fichiers individuels suivants. Je ne comprends pas pourquoi ils ne sont pas tous regroupés dans une seule archive :

adafruit_bmp280.mpy
adafruit_clue.mpy
adafruit_lis3mdl.mpy
adafruit_sht31d.mpy
adafruit_slideshow.mpy
neopixel.mpy

Exemple de code

Pour un premier essai simple avec l'environnement Python, vous pouvez utiliser l'exemple fourni sur <https://learn.adafruit.com/adafruit-clue/clue-spirit-level>. Il s'agit d'une application de niveau à bulle utilisant plusieurs idiomes spécifiques à CLUE.

La première étape du code consiste à inclure un ensemble de bibliothèques :

```
import board
import displayio
from adafruit_display_shapes.circle import Circle
from adafruit_clue import clue
```



Figure 6. Un niveau à bulle utilisant les capteurs intégrés.

Publicité

TAILORED TO YOUR NEEDS.

Custom & Standard Terminal Blocks



WÜRTH
ELEKTRONIK
MORE THAN
YOU EXPECT



Würth Elektronik Terminal Blocks

In addition to a portfolio of more than 2000 standard articles, Würth Elektronik offers various possibilities to tailor the products to your specific requirements. Personalized modifications of standard terminal blocks are available for small to medium quantities within a few days as a special service. Fully customized products in high quantities are possible within a few weeks. In house design, tooling and prototyping ensures all customer specific requirements are met.

For further information, please visit: www.we-online.com/TBL

- Highly customized products
- Over 2000 standard articles
- Available from stock without MOQ
- Fast delivery
- Personalized modifications of standard parts for small quantities
- Color & printing possibilities with MOQ for mass production

En plus de l'objet `clue`, qui assure diverses fonctions liées à la carte, importer la classe `Circle` est également intéressant ici. La pile GUI permet à la fois de dessiner directement dans un *frame buffer* et de travailler avec des objets, convertis par le micrologiciel, en éléments visibles à l'écran.

Dans la section suivante, le micrologiciel initialise une référence à l'écran et assemble un objet de type groupe pour l'écran :

```
display = board.DISPLAY
clue_group = displayio.Group(max_size=4)
```

L'objet `clue_group` est intéressant parce qu'il crée un élément parent qui rappelle un arbre DOM. Notre code crée donc plus ou moins des objets arbitraires dans cet arbre à afficher.

Regardez la photo de la **figure 6**, la prochaine action du programme est de créer les trois cercles qui représentent la déviation et à les enregistrer pour la sortie :

```
outer_circle = Circle(120, 120, 119, outline=clue.WHITE)
middle_circle = Circle(120, 120, 75, outline=clue.
    YELLOW)
inner_circle = Circle(120, 120, 35, outline=clue.GREEN)
clue_group.append(outer_circle)
clue_group.append(middle_circle)
clue_group.append(inner_circle)
```

On trouve ensuite quelques tâches de « ménage », dont le sens est plus facile à comprendre en examinant l'exemple de code ci-dessous :

```
x, y, _ = clue.acceleration
bubble_group = displayio.Group(max_size=1)
level_bubble = Circle(int(x + 120), int(y + 120), 20,
    fill=clue.RED, outline=clue.RED)
bubble_group.append(level_bubble)

clue_group.append(bubble_group)
display.show(clue_group)
```

Enfin et surtout, nous avons besoin d'une boucle qui analyse les valeurs de position fournies par la bibliothèque Adafruit via l'attribut `acceleration` et qui les écrit dans les propriétés de l'objet `bubble_group` :

```
while True:
    x, y, _ = clue.acceleration
    bubble_group.x = int(x * -10)
    bubble_group.y = int(y * -10)
```

La façon la plus pratique d'exécuter rapidement du code sur la carte CLUE est d'utiliser le fichier *code.py* de la figure 5. Le micrologiciel CircuitPython l'exécute automatiquement à chaque démarrage. La figure 6 montre ce que vous pouvez obtenir.

Comme la carte est dotée du Bluetooth, ce peut être un canal pour communiquer avec l'ordinateur hôte. Sur [3], Adafruit fournit un exemple amusant qui illustre l'utilisation de l'API web Bluetooth implémentée dans Google Chrome.

Et maintenant en C

Le langage Python permet d'obtenir rapidement des résultats « non bureaucratiques » avec un système embarqué. Cependant, la puissance maximale est atteinte avec le langage C. Sur un système monocœur avec liaison radio, la mise en œuvre de la communication est un défi si l'on veut que l'application ne souffre pas de problèmes de synchronisation. C'est pourquoi Adafruit oblige plus ou moins les développeurs à utiliser l'EDI Arduino. Un système d'exploitation en temps réel fonctionne alors en arrière-plan, allouant la puissance de calcul aux différentes tâches.

Sous Linux, la première étape nécessite un paquet d'extension qui permet à l'EDI Arduino (version 1.8.6 ou supérieure) de communiquer avec le *bootloader* non standard de CLUE :

```
tamhan@TAMHAN18:~$ pip3 install --user adafruit-nrfutil
Collecting adafruit-nrfutil
...
Successfully installed adafruit-nrfutil-0.5.3.post13
```

Ensuite, il faut saisir l'URL www.adafruit.com/package_adafruit_index.json dans le *Gestionnaire de carte* pour rendre le package de la carte *Adafruit nRF52* disponible pour le téléchargement. Une fois ces étapes terminées, la carte sera disponible sous *Outils > Type de carte > Adafruit CLUE*.

Malheureusement, comme dans le cas de CircuitPython, l'installation de ces bibliothèques et des autres paramètres est un processus laborieux. Vous trouverez plus d'informations à ce sujet dans [4].

LIENS

[1] StreamIT, s. r. o. : <https://edutronik.sk/v2/>

[2] Boîtier : www.thingiverse.com/thing:1767446

[3] Appli de démonstration : <https://learn.adafruit.com/bluefruit-dashboard-web-bluetooth-chrome>

[4] Informations : <https://learn.adafruit.com/adafruit-clue?view=all>

Carte CLUE : vaut-elle le coup ?

Avec la carte CLUE, vous disposez d'une plateforme d'évaluation très attrayante, agréable à utiliser dans le domaine de l'interfaçage et en plus dotée d'un écran couleur. Toutefois son prix est relativement élevé par rapport à la carte BBC micro:bit (nettement moins chère). Notez également que la carte CLUE n'est pas compatible à 100% avec micro:bit. L'expérience nous apprend que ce détail « insignifiant » a un fort impact, surtout lorsque l'on se trouve dans une situation difficile, comme la migration d'un projet existant et fonctionnel entre les deux plateformes.

Pour ceux qui veulent travailler avec un microcontrôleur « pur », basé sur la technologie Nordic ou tout autre module radio, vous serez probablement mieux servis par une carte d'évaluation classique. En résumé, la carte CLUE est un produit attachant pour ceux qui apprécient la carte BBC micro:bit, mais qui ont besoin d'un peu plus de puissance ou d'un véritable écran. ◀

210395-04

Des questions, des commentaires ?

Contactez Elektor (redaction@elektor.fr).

Contributeurs

Texte et illustrations : Tam Hanna

Rédaction : Jens Nickel

Traduction : Asma Adhimi

Mise en page : Giel Dols



PRODUITS

- Carte CLUE d'Adafruit
www.elektor.fr/19512

- Livre « Initiation au langage CircuitPython et à la puce nRF52840 », Michaël Bottin, Elektor, ISBN 978-2-86661-211-5
www.elektor.fr/19523



NOTRE GAMME

PAR DES TECHNICIENS POUR LES TECHNICIENS

The best part of your project:
www.reichelt.com

Uniquement le meilleur pour vous - provenant de plus de 900 marques

Nos responsables produits sont employés par Reichelt depuis de nombreuses années et connaissent les exigences de nos clients. Ils rassemblent une large gamme de produits de qualité, à la fois parfaits pour les besoins dans les domaines de la recherche et du développement, la maintenance, l'infrastructure informatique et la production en petites séries et adaptés pour les fabricants.

Arduino chez reichelt



Le TinkerKit Braccio est un bras robotique entièrement fonctionnel contrôlé par Arduino. Il peut être assemblé de différentes manières pour différentes tâches telles que le déplacement d'objets.

- SpringRC SR431 - Servo à double sortie
- Portée maximale : 80 cm
- Hauteur maximale : 52 cm
- Capacité de charge : poids maximal à 32 cm de portée : 150 g

N° de commande.:
ARD TINKER BOT

236,96
(197,47)



Découvrez maintenant ► www.reichelt.com/arduino

Types de paiement :



PRIX DU JOUR ! Prix à la date du: 15. 2. 2022

- Excellent rapport qualité prix
- Plus de 120 000 produits sélectionnés
- Livraison fiable - depuis l'Allemagne dans le monde entier

reichelt
elektronik – Tirer le meilleur parti de votre projet

www.reichelt.com

Assistance téléphonique: +33 97 518 03 04

Les réglementations légales en matière de résiliation sont applicables. Tous les prix sont indiqués en € TVA légale incluse, frais d'envoi pour l'ensemble du panier en sus. Seules nos CGV sont applicables (sur le site <https://rcl.it/CG-FR> ou sur demande). Semblables aux illustrations. Sous réserve de coquilles, d'erreurs et de modifications de prix. reichelt elektronik GmbH & Co. KG, Elektronikring 1, 26452 Sande (Allemagne), tél. +33 97 518 03 04