


Quelques idées

Auteur : Christophe LIGERET



CircuitPython Starter Kit with Adafruit Itsy Bitsy M4

PRODUCT ID: 4028

\$24.95

OUT OF STOCK

Please enter your details below and we will send you an email when this item is back in stock. You will only be emailed about this product!

YOUR NAME

YOUR EMAIL

[NOTIFY ME](#)

[ADD TO WISHLIST](#)

[DESCRIPTION](#)



Carte micro:bit

Code article : 35340

La carte micro:bit est un nano-ordinateur créé pour la promotion de la programmation dans l'éducation.

[Description complète](#)

Quantité : [+](#) [-](#)

✓ Quantité en stock : 422

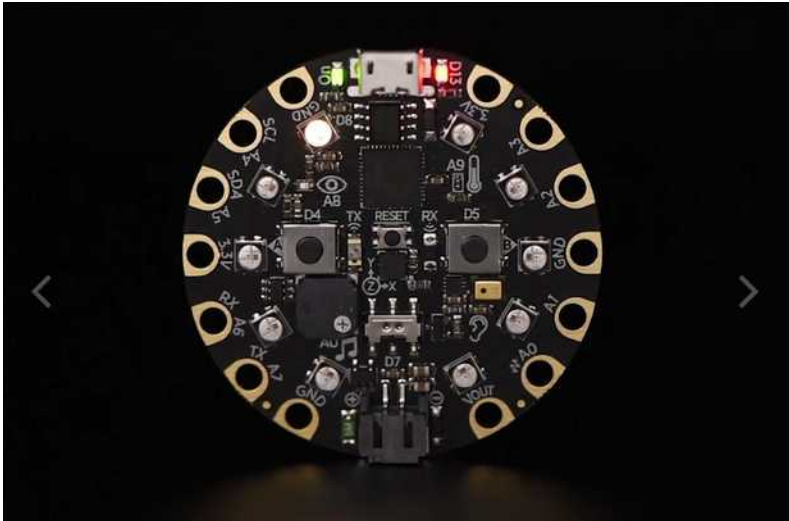
Livraison à partir de 2,90€ ?

16,58 € HT

19,90 € TTC

dont 0,01 € d'éco-part

[Ajouter au panier](#)



Circuit Playground Express

PRODUCT ID: 3333

\$24.95

IN STOCK

[ADD TO CART](#)

☐ Also include 1 x Adafruit Circuit Playground Express Enclosure (\$4.95)

☐ Also include 1 x Bolt-On Kit for Circuit Playground, micro:bit, Flora or Gemma (\$1.50)

☐ Also include 1 x Learn CircuitPython with 1 Month Subscription to Codecademy Pro (\$19.99)

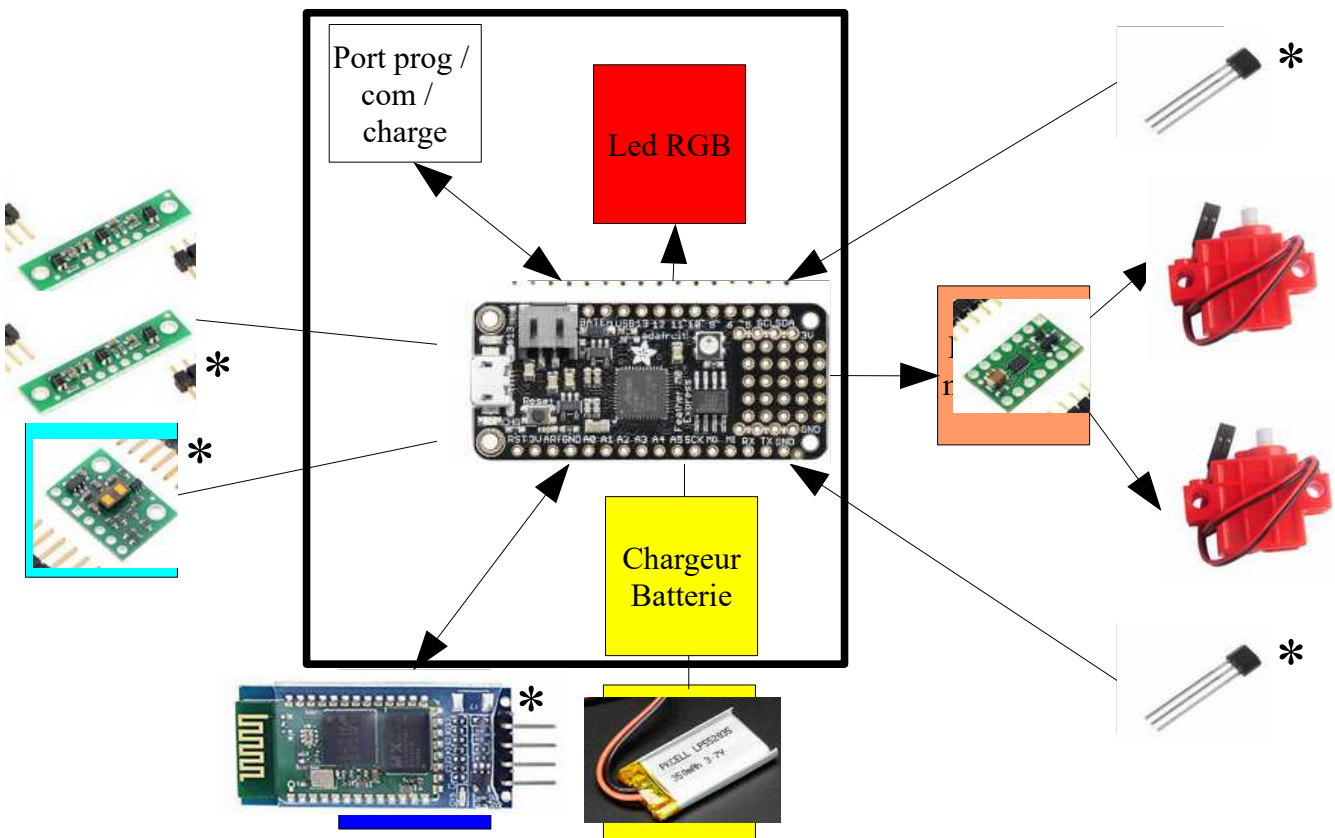
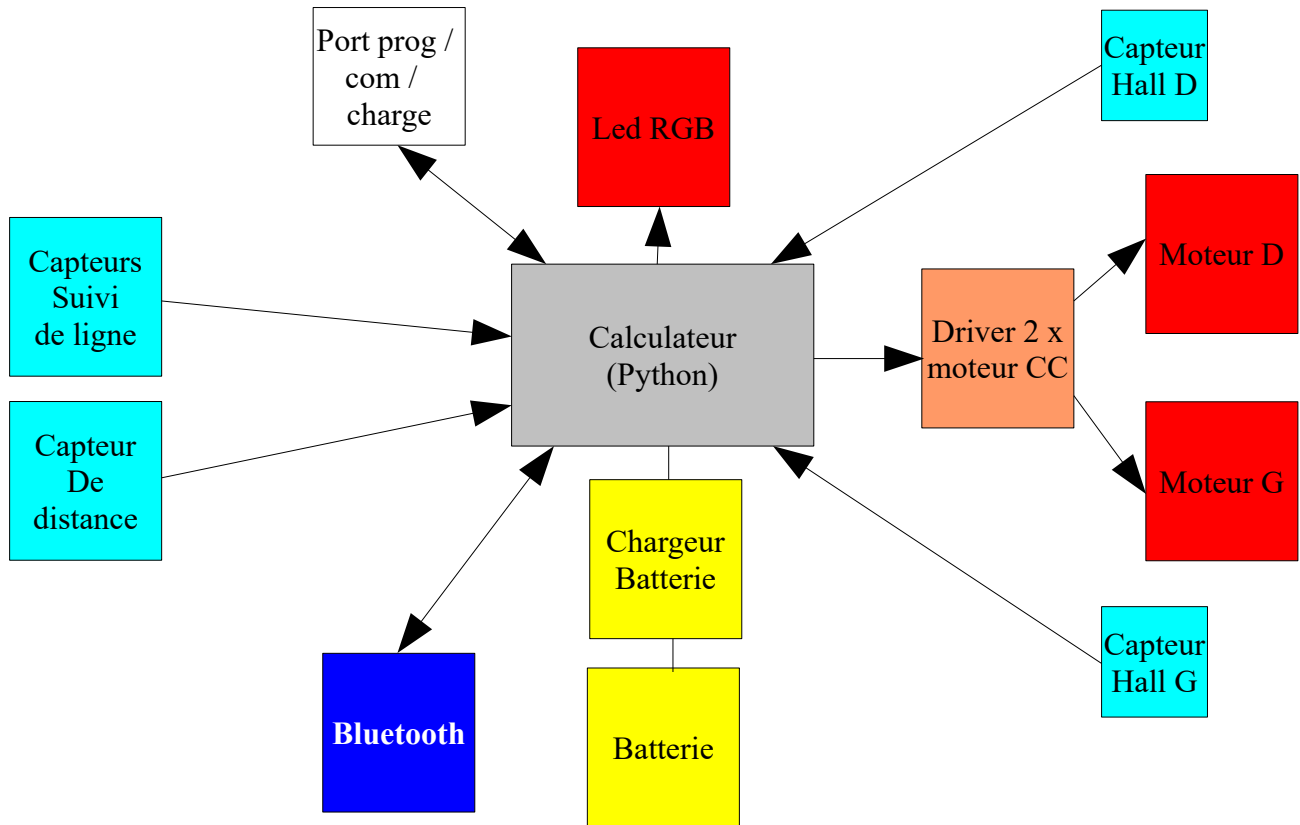
QTY	DISCOUNT
1-9	\$24.95
10-99	\$22.46
100+	\$19.96

[ADD TO WISHLIST](#)

[DESCRIPTION](#)

[TECHNICAL DETAILS](#)

Architecture du robot

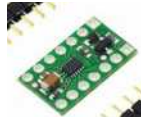


Liste du matériel

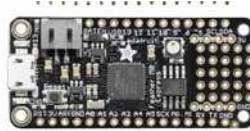
2 x Red Gear Motor for Geek Servo (2 x 3,28 € sur aliexpress)



1 x DRV8835 2x1,2A (4,90 € sur Gotronic)



1 x Feather M0 express (17,70 € sur Adafruit)



1 ou 2 x capteur de ligne (4,55 € sur Gotronic)



1 x module bluetooth (~ 5 €)



1 x capteur de distance matriciel ToF avec ROI (VL53L1X : 11,95 € sur Gotronic)



1 x paire d'encodeurs magnétiques (3081 : 9,95 € sur Gotronic)



ou bien



2 ou 4 x capteur à effet Hall alimenté en 3,3V (~ 1 € l'unité)



1 x batterie Lithium Ion 3,7V / 350 mAh (6,16 € sur Adafruit)



Coût total du matériel :

Robot de base : 40 €

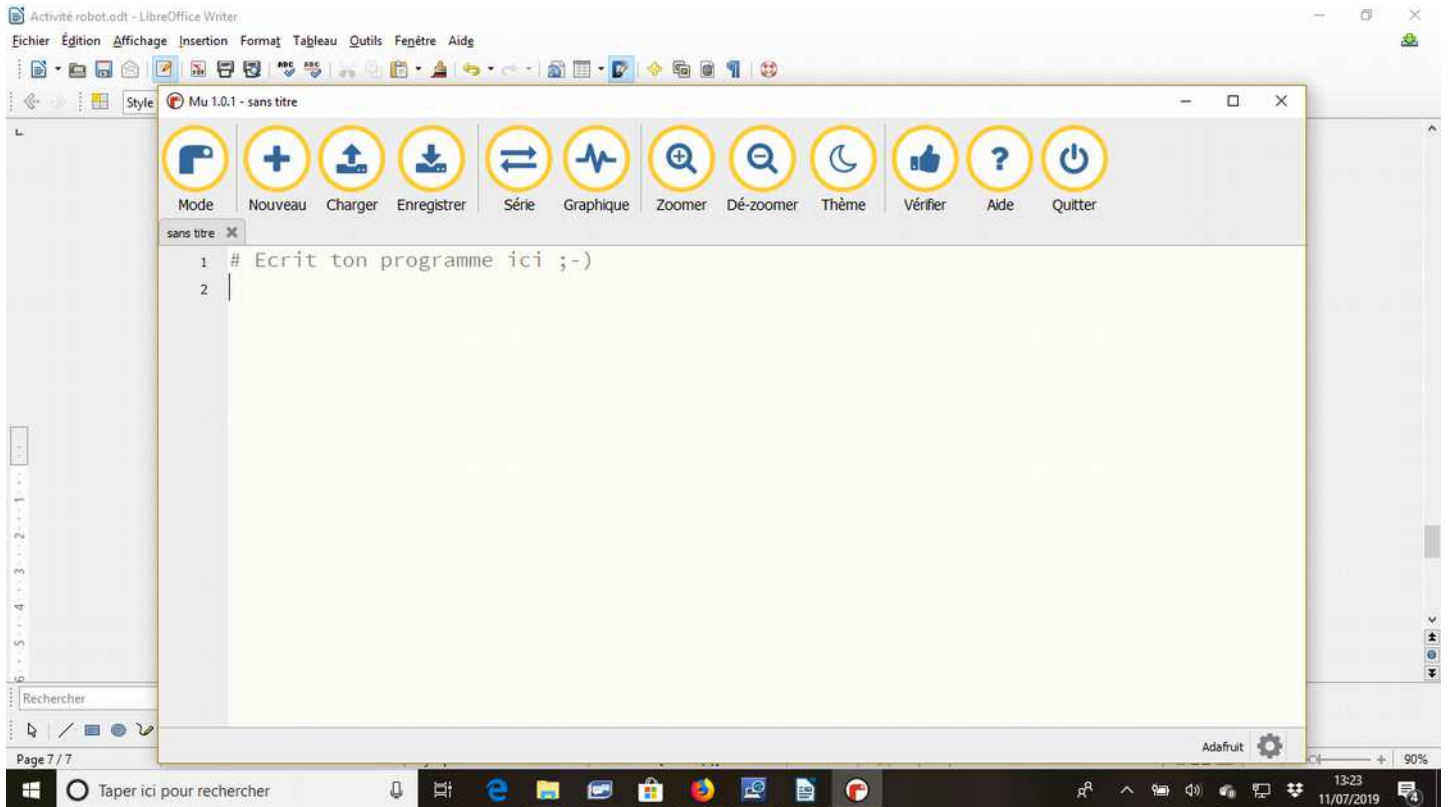
codeurs incrémentaux : +5 €

communication bluetooth : +5 €

télémètre avant : +10 €



IDE : Mu editor



Activité 1 : mesure de l'éclairage ambiant

matériel nécessaire : robot dans sa version de base

durée : 1 - 2h

Objectif : à partir de la mesure de l'intensité lumineuse captée par les 6 capteurs infrarouges, piloter la couleur de la led RVB intégrée au calculateur

Partie I : mesure de l'intensité lumineuse

1°) Réaliser en langage Python un programme qui lit et place dans un tableau les intensités reçus par chacun des 6 capteurs. Ces 6 données seront stockées dans la liste **tableau_capteurs**.

On initialisera : **tableau_capteurs = [0 , 0 , 0 , 0 , 0 , 0]**

La période d'acquisition sera de 1 seconde

Partir du code ci-dessous et afficher les mesures dans la console de Mu Editor

```
import time
```

```
import board
```

```
from analogio import AnalogIn
```

```
capteur_IR_A0 = AnalogIn(board.A0)
```

```
tableau_capteurs = [ 0 , 0 , 0 , 0 , 0 , 0 ]
```

```
time.sleep(1)
```

2°) Effectuer un calcul statistique sur les 6 données :

- moyenne des 6 valeurs

- moyenne de cette moyenne dans le temps : 10 échantillons : on affiche une moyenne toutes les 10 secondes

Partie II : pilotage de la couleur de la led RVB

Ecrire en langage Python un programme qui pilote la couleur de la led en fonction de l'intensité de l'éclairage ambiant.

1°) Pilotage en tout-ou-rien avec ou sans hystérésis ou temporisations

Utiliser votre robot comme un lampadaire automatique.

2°) Pilotage continu des couleurs : aller de manière linéaire du bleu (obscurité) au jaune (intensité maximale).

Penser à utiliser des fonctions affines pour commander chacune des composantes rouge, verte et bleu.

En coordonnées (R ; V ; B) :

Obscurité : (0 ; 0 ; 255)

Eclairage ambiant maximum : (255 ; 255 ; 0)

Partie III : pilotage de la couleur de la led RVB ainsi que de son clignotement

Ecrire en langage Python un programme qui pilote la couleur de la led en fonction de l'intensité de l'éclairage ambiant (cf. Partie II) et qui en plus clignote si les 6 valeurs mesurées sont très dispersées, ou reste fixe si les valeurs sont peu dispersées..

Activité 2 : suivi automatique de ligne

matériel nécessaire : robot dans sa version de base

durée : 2 - 4h

Objectif : faire suivre au robot une ligne

Partie I : mesure de la réflectance du sol

1°) Réaliser un programme Python qui lit les 6 capteurs infrarouges selon que les leds IR sont activées ou non et faire la différence des mesures avec leds IR activées / non activées.

Selon vous, pourquoi fait-on cette différence ?

On utilisera les listes : **tableau_IR_High** ; **tableau_IR_Low** et **tableau_IR_Diff**

2°) Réaliser un programme qui renormalise les valeurs entre 0 et 1 selon la valeur minimum et maximum des 6 mesures. On souhaite que les valeurs soient maximales sur la ligne et minimales sur le sol.

On utilisera la liste **tableau_IR_Norm**

Partie II : calcul d'un indicateur de position

1°) Ecrire un programme qui considère les valeurs contenues dans la liste **tableau_IR_Norm** comme une série statistique dont on veut déterminer la médiane (indicateur de position)

a) Dans la liste **tableau_IR_ECC**, stocker les effectifs cumulés croissants.

b) Dans la liste **tableau_IR_FCC**, stocker les fréquences cumulés croissantes.

c) Déterminer l'indice **indice** tel que **tableau_IR_FCC[indice-1] < 0.5 <= tableau_IR_FCC[indice]**

En déduire un encadrement de la médiane.

d) En utilisant une interpolation affine entre les points (**indice-1** ; **tableau_IR_FCC[indice-1]**) et (**indice** ; **tableau_IR_FCC[indice]**), donner une première estimation de la médiane

e) En utilisant une technique des barycentres des points (**indice-1** ; **tableau_IR_FCC[indice-1]**) et (**indice** ; **tableau_IR_FCC[indice]**) (faire une recherche), donner une seconde estimation de la médiane

2°) Faire des tests et évaluer chacune de ces deux estimations.

Partie III : asservissement du robot sur la ligne

Idéalement, la médiane est comprise entre 0 et 5, ainsi, lorsque le robot est parfaitement centré sur la ligne, la médiane devrait être égale à 2,5. En pratique, ce n'est jamais le cas.

1°) Ecrire en langage Python un programme qui calcule l'erreur d'asservissement du robot

erreur = consigne - mesure

2°) Ecrire deux fonctions informatiques qui pilotent respectivement le moteur A et le moteur B. Chacun prend en argument la vitesse comprise entre -65535 et 65535

def commandeMoteurA(vitesse):

def commandeMoteurB(vitesse):

3°) Ecrire les instructions permettant au robot de se recentrer sur la ligne chaque fois qu'il s'en écarte.

4°) En reprenant les données présentes dans la liste **tableau_IR_Diff**, déterminer le plus grand écart entre ces valeurs (on l'appellera contraste) et après avoir fixé un seuil expérimental, commander l'arrêt du robot lorsqu'il a perdu la ligne.

Activité 3 : contrôle à distance du robot

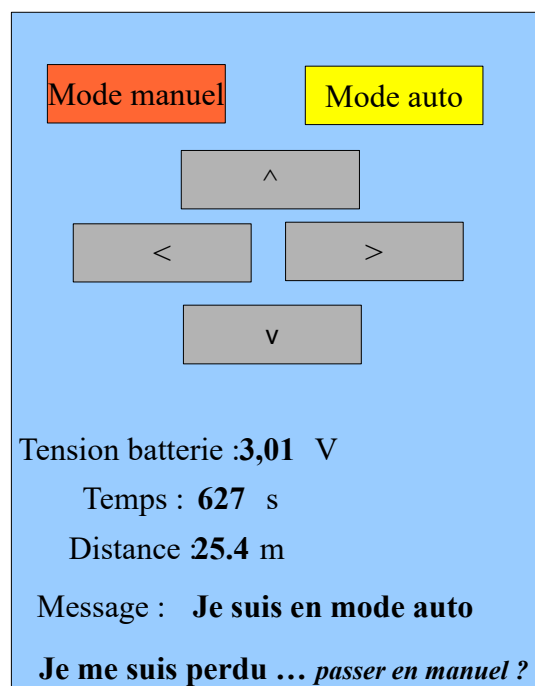
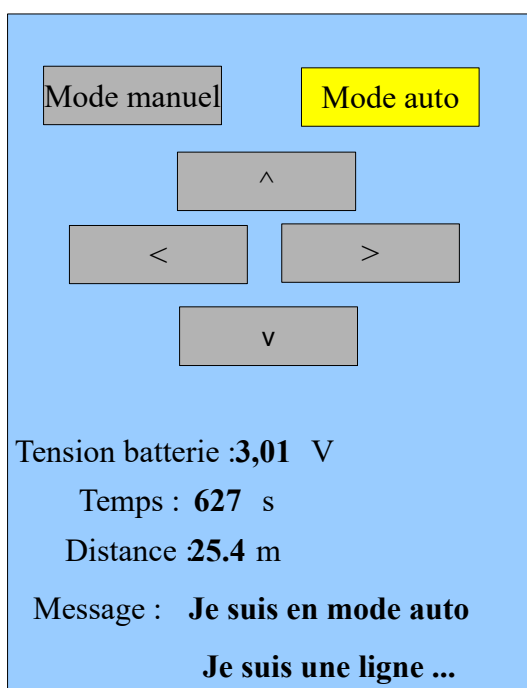
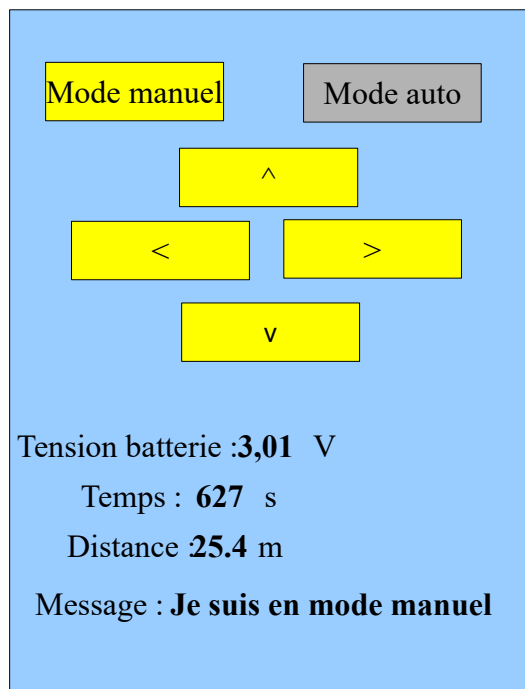
matériel nécessaire : robot dans sa version base + bluetooth

durée : 2 - 4h

Objectif : piloter à distance le robot et accéder à son état

En utilisant l'application Processing, développer un programme qui permette de communiquer avec le robot :

- lui envoyer des commandes sur ses moteurs
- lui demande de suivre une ligne en mode automatique
- le robot répond qu'il s'est perdu s'il ne voit plus la ligne
- le robot indique la température de son processeur ; la tension de la batterie ; le nombre de minutes écoulées depuis la dernière mise sous tension : une estimation (grossière) du nombre de mètres parcourus depuis la dernière mise sous tension.



Pour les plus bricoleurs



Ecrans TFT



Ecrans OLED



Ecrans alphanumériques



Boutons poussoirs



Boutons lumineux



Potentiomètres



Voyants leds



Cadrans



Joystick



Buzzers