

Kit IoT CoreS3 K128

Code : 38353

M5stack

Kit combinant un module CoreS3 et un module d'extension DinBase. Intègre des circuits de mesures spatiales, un circuit RTC, 2 micros, un amplificateur I2S et un HP.

59,92 €_{HT}

71,90 €_{TTC}

dont 0,02 € d'éco-part

Description

Présentation et fonctionnalités :

Kit combinant un module CoreS3 de 3ème génération de la série M5Stack Core et un module d'extension DinBase.

Cette version CoreS3 est basée sur un ESP32-S3 à 240 MHz avec interface WiFi et afficheur IPS tactile offrant une excellente luminosité et un fort contraste.

L'extension DinBase comporte une batterie, une petite zone de prototypage, un circuit d'alimentation et de charge.

Le CoreS3 est également doté :

- d'un capteur de proximité
- d'une caméra frontale 0,3 MPx
- de 2 micros
- d'un circuit IMU 6 axes (accéléromètre et gyroscope) et d'un magnétomètre
- d'un amplificateur audio I2S et d'un haut-parleur 1 W
- d'une embase *PortA* : Vcc, GND, broches 8 et 9

Le module DinBase complète le CoreS3 avec plusieurs fonctions :

- alimentation via une source 9 à 24 Vcc
- alimentation via un accu LiPo 3,7 Vcc/500 mAh intégré (charge via USB)
- la fixation sur rail DIN ou murale
- une embase *PortB* : Vcc, GND, broches 8 et 9
- une embase *PortC* : Vcc, GND, broches 17 et 18

Ce kit est un choix idéal pour les projets connectés polyvalents à faible consommation, pour le prototypage, les systèmes domotiques et l'automatisme industriel.

Programmation et communication :

La carte CoreS3 se programme depuis son port USB Type-C (cordon à prévoir) avec :

- l'IDE en blocs et en Python [UIFlow](#) : pour réaliser une interface graphique et utiliser facilement les fonctions disponibles
- l'IDE [Arduino](#) : pour les utilisateurs avancés (exemple disponible sur le [github du fabricant](#))

Connectique :

- port USB Type-C compatible OTG pour la programmation, l'alimentation et la charge de l'accu LiPo.
- lecteur pour carte microSD (non incluse) pour le stockage de fichiers multimédia ou des données mesurées (horodatage possible via horloge temps réel intégré).

- port GPIO du module CoreS3 pour enficher l'extention Din Base.

Contenu :

- 1 x module M5Stack CoreS3
- 1 x module d'extension DinBase
- 1 x clé 6-pans

Remarques :

Le cordon USB Type-C de programmation est à prévoir séparément.

Les E/S de ce module sont uniquement compatibles 3,3 V. Une tension supérieure endommagerait le module.

Pour des raisons de réglementation aérienne, ce produit ne peut pas être exporté.

Caractéristiques :

- Alimentation :
 - 5 Vcc via le connecteur USB Type-C (cordon non inclus)
 - 9 à 24 Vcc via le connecteur d'alimentation 5,5 x 2,1 mm (alim. non incluse)
 - via la batterie LiPo 3,7 Vcc/500 mAh intégrée
 - charge de la batterie via l'USB
- Microcontrôleur ESP32-S3 :
 - microprocesseur : Xtensa LX7 dual-core à 240 MHz
 - mémoire Flash : 16 MB
 - mémoire PSRAM : 8 MB
- Afficheur :
 - technologie : IPS
 - dimensions : 2"
 - résolution : 320 x 240 pixels
 - driver : ILI9342C
 - tactile capacitif
- Haut-parleur 1 W (amplificateur I2S)
- 2 x micros
- Caméra GC308 0,3 MPx
- Capteur de proximité LTR-553ALS-WA
- Circuit IMU 6 axes BMI270 :
 - accéléromètre : ± 2 g - ± 4 g - ± 8 et ± 16 g
 - gyroscope : ± 125 °/s - ± 250 °/s - ± 500 °/s - ± 1000 °/s - ± 2000 °/s
- Magnétomètre BMM150 :
 - axes x et y : ± 1300 μ T
 - axe z : ± 2500 μ T
- Lecteur microSD (16 GB maxi - carte non incluse)
- Circuit RTC BM8563 (horloge temps réel)
- Interfaces GPIO (CoreS3 + DinBase) :
 - embase HY2.0-4P : PortA (Vcc, GND, broches 8 et 9 de l'ESP32)
 - embase HY2.0-4P : PortB (Vcc, GND, broches 8 et 9 de l'ESP32)
 - embase HY2.0-4P : PortC (Vcc, GND, broches 17 et 18 de l'ESP32)
- Inverseur ON-OFF sur DinBase
- Bouton-poussoir ON-OFF sur CoreS3
- Bouton-poussoir reset
- Dimensions : 54 x 54 x 16 mm
- Poids : 73,3 g

Ressources

- [Fiche technique](#)
- [Fiche technique ESP32-S3](#)
- [Fiche technique BMI270](#) (IMU 6 axes)
- [Fiche technique BMM150](#) (magnétomètre)
- [Fiche technique caméra GC0308](#)
- [Schéma](#)
- [Exemple de code Arduino](#)
- [Librairie Arduino pour BMI270](#)
- [IDE UIFlow](#)
- [IDE Arduino](#)