Bienvenue sur Microcontrôleurs

Rédacteur(s) : Philippe Mariano

[Mise à jour le 16/6/2024] En cours de réorganisation

Présentation

Ce Wiki est consacré à la **mise en oeuvre** et à la **programmation** de cartes à **microcontrôleur** (Arduino, ESP, Raspberry Pi Pico, etc.:
- en **microPython** ou **circuitPython** (élèves ayant la spécialité **NSI**)
- en **C**, C++, **Arduino** (élèves ayant la spécialité **SIN**)

MicroPython est une implémentation simple et efficace du langage de programmation Python 3, qui inclut un petit sous-ensemble de la bibliothèque standard Python et qui est optimisée pour fonctionner sur des microcontrôleurs. Il est suffisamment compact pour s’adapter à 256 ko d’espace de code et à 16 ko de RAM.

CircuitPython est un dérivé open source du langage de programmation MicroPython destiné aux étudiants et aux débutants. Le développement de CircuitPython est soutenu par Adafruit Industries. Il s'agit d'une implémentation logicielle du langage de programmation Python 3, écrit en C.

Sommaire

1. RANDOM NERD TUTORIALS[1)]
2. **NSI** - "Etape par étape" - Premiers programmes en MicroPython ou CircuitPython avec une carte Raspbery Pi Pico
3. **SIN** - "Etape par étape" - Premier programme avec une carte Arduino et l'IDE v2

2. **MATERIELS**
    1. **Cartes de prototypage à microcontrôleurs**
        1. Arduino Uno (Wifi), Mega 2560
        2. Arduino MKR Wifi 1010
        3. Arduino GIGA R1 Wifi
        4. Raspberry Pi Pico et Pico Wifi
        5. Les modules Espressif
            1. ESP01(S)
            2. ESP8266
            3. ESP32
        6. BrainPad de GHI Electronics | MakeCode
        7. Micro:bit | Documentation
        8. Seeed Wio Terminal | PasswordVault on Seeed Studio Wio Terminal
        9. Adafruit
            1. Pyportal
            2. Metro M4
        10. Pycom
        11. Particles
    2. **Autres matériels : capteurs, afficheur, préactionneurs, actionneurs, etc.**
3. **LOGICIELS**
    1. **Simulateurs**
        1. Wokwi pour Arduino Uno, STM32, ESP32 et Raspberry Pi Pico
        2. Makecode pour la carte BrainPad de GHI Electronics
        3. MicroPython sur Unicorn
    2. **IDE : Environnements de Développement Intégré**
        1. Thonny (Python, MicroPython, CircuitPython) | To begin
        2. Arduino IDE v2 (C, C++) | Utilisez l'IDE Arduino V2 avec un NAS Synology
    3. **Mise en oeuvre des périphériques du microcontrôleur**
        1. Communication
            1. BUS (RS232, I²C, SPI)
            2. Réseaux
        2. Entrées, sorties (GPIO)
            1. Généralités
            2. Entrées, sorties numériques
            3. Entrées analogiques
        3. Gestion du temps
            1. Timer
            2. Horloge Temps Réel
            3. WatchDog
                1. Référence Arduino
                2. Watchdog Arduino : explication du fonctionnement, et exemples de code
        4. Stockage
            1. Le système de fichiers LittleFS (évolution de SPIFFS) des ESP8266 et ESP32
    4. **IoT - Objets connectés**
        1. MQTT
            1. Mise en oeuvre d'un client MQTT sur un EP8266 (ESP32) Feather Huzzah ou

        un MKR Wifi 1010
    2. Clouds
        1. Cloud Arduino
        2. ESP RAINMAKER
5. **Bases De Données**
    1. BDDR - Bases de données relationnelles et SGBDR (Sommaire)
    2. Firebase - “Control ESP32/ESP8266 GPIOs from Anywhere” : [Article 1] [Article 2 : Web App]
6. **Multitâche**
    1. Introduction to RTOS
    2. Write non-blocking code
    3. Scheduler library
    4. Azure RTOS ThreadX for Arduino 101: Threads
    5. Discussions pour un “vrai” multitâche Introducing multitasking to Arduino
7. **Web**
    1. Serveur HTTP utilisé en projet
    2. Arduino Uno - PHP - MySQL
    3. ESP8266 First Web Server [doc]
4. **SAUVEGARDE ET COLLABORATION**
    1. Transférer des fichiers avec FileZilla client
    2. Gestion de versions : démarrer avec Git et Github
    3. Travail collaboratif dans VSCode
    4. Compte utilisateur sur le NAS Synology
    5. Faire fonctionner l'IDE Arduino V2 sur un NAS Synology
5. **RESSOURCES TECHNIQUES**
    1. Ultimate Guide to Switch Debounce
    2. From Zero to main(): How to Write a Bootloader from Scratch
    3. Débogage des cartes Arduino® basées sur SAM avec Atmel-ICE
    4. **Assembleur**
        1. Articles : Pourquoi programmer en assembleur ?
        2. Module 1 (vidéo) : Microcontrôleur Atmega 328P et sa programmation en langage assembleur

- **WEBOGRAPHIE (Bibliothèques, tutoriels)**
    - Arduino
- **BIBLIOGRAPHIE**
    - Arduino

1)

Random Nerd Tutorials helps makers, hobbyists and engineers build electronics projects. We make projects with: ESP32, ESP8266, Arduino, Raspberry Pi, ...