

ESP8266Sketch2_InfoSysteme

Ce code pour l'ESP8266 permet d'afficher des informations détaillées sur le microcontrôleur et sa mémoire, via une connexion série à une vitesse de 115200 bauds.

Il utilise des fonctions spécifiques à l'ESP8266 fournies par la bibliothèque **ESP** pour récupérer et afficher ces informations.

```
#include <Arduino.h>

ADC_MODE(ADC_VCC);

void setup()
{
  Serial.begin(115200);
}

void loop()
{
  Serial.println("INFO ESP8266EX");
  Serial.printf("Numéro de série de l'ESP8266EX .....: %u\n", ESP.getChipId());

  float frequenceCpu = float(ESP.getCpuFreqMHz());
  Serial.printf("Fréquence du CPU .....: %4.1f MHz\n", frequenceCpu);

  float vcc = float(ESP.getVcc()) / 1000;
  Serial.printf("Alimentation .....: %4.2f V\n", vcc);

  String chaine = ESP.getCoreVersion();
  chaine.replace('_', '.');
  Serial.print("Version du gestionnaire de carte...: ");
  Serial.println(chaine);

  Serial.printf("Version du SDK .....: %s\n\n", ESP.getSdkVersion());
  Serial.println("INFO FLASH");
  Serial.printf("Numéro de série du chip .....: %u\n", ESP.getFlashChipId());

  float frequenceFlash = float(ESP.getFlashChipSpeed()) / 1000000;
  Serial.printf("Fréquence .....: %4.1f MHz\n", frequenceFlash);
  Serial.printf("capacité effective .....: %u octets\n", ESP.getFlashChipRealSize());
  Serial.printf("capacité paramétrée dans l'EDI .....: %u octets\n", ESP.getFlashChipSize());
  Serial.printf("Taille du sketch .....: %u octets\n", ESP.getSketchSize());
  Serial.printf("Mémoire disponible .....: %u octets\n\n\n", ESP.getFreeSketchSpace());
  delay(100000);
}
```

```
ADC_MODE(ADC_VCC);
```

- Configure l'ESP8266 pour utiliser l'ADC (Analog Digital Converter) en mode **ADC_VCC**, ce qui permet de lire la tension d'alimentation de l'ESP8266 via la broche ADC.

```
Serial.println("INFO ESP8266EX");  
Serial.printf("Numéro de série de l'ESP8266EX .....: %u\n", ESP.getChipId());
```

- Renvoie l'identifiant unique (numéro de série) de l'ESP8266, qui est un entier non signé (32 bits). Cela permet d'identifier de manière unique chaque module ESP8266.

```
float frequenceCpu = float(ESP.getCpuFreqMHz());  
Serial.printf("Fréquence du CPU .....: %4.1f MHz\n", frequenceCpu);
```

- Retourne la fréquence du processeur de l'ESP8266 en MHz (généralement 80 MHz ou 160 MHz).
- La fréquence est stockée dans une variable float et affichée avec une précision d'une décimale.

```
float vcc = float(ESP.getVcc()) / 1000;  
Serial.printf("Alimentation .....: %4.2f V\n", vcc);
```

- Mesure la tension d'alimentation de l'ESP8266 en millivolts. En divisant par 1000, on obtient la valeur en volts avec une précision de deux décimales.

```
String chaine = ESP.getCoreVersion();  
chaine.replace('_', '.');  
Serial.print("Version du gestionnaire de carte...: ");  
Serial.println(chaine);
```

- Renvoie la version du noyau (gestionnaire de carte) de l'ESP8266 sous forme de chaîne.
- Le remplacement des underscores par des points dans la version du noyau est effectué pour améliorer la lisibilité.

```
Serial.printf("Version du SDK .....: %s\n\n", ESP.getSdkVersion());
```

- Renvoie la version du SDK (Software Development Kit) utilisé par le module ESP8266.

```

Serial.printf("Version du SDK .....: %s\n\n", ESP.getSdkVersion());
Serial.println("INFO FLASH");
Serial.printf("Numéro de série du chip .....: %u\n", ESP.getFlashChipId());
float frequenceFlash = float(ESP.getFlashChipSpeed()) / 1000000;
Serial.printf("Fréquence .....: %4.1f MHz\n", frequenceFlash);
Serial.printf("capacité effective .....: %u octets\n", ESP.getFlashChipRealSize());
Serial.printf("capacité paramétrée dans l'EDI .....: %u octets\n", ESP.getFlashChipSize());
Serial.printf("Taille du sketch .....: %u octets\n", ESP.getSketchSize());
Serial.printf("Mémoire disponible .....: %u octets\n\n\n", ESP.getFreeSketchSpace());

```

- **ESP.getFlashChipId()** : Renvoie l'identifiant unique de la puce Flash utilisée sur l'ESP8266.
- **ESP.getFlashChipSpeed()** : Retourne la fréquence de la mémoire Flash en Hz. La valeur est divisée par 1 000 000 pour l'obtenir en MHz.
- **ESP.getFlashChipRealSize()** : Renvoie la taille réelle de la mémoire Flash en octets.
- **ESP.getFlashChipSize()** : Renvoie la taille de la mémoire Flash configurée dans l'EDI (environnement de développement intégré), souvent paramétrée à une taille différente de la taille réelle.
- **ESP.getSketchSize()** : Indique la taille actuelle du programme (sketch) chargé sur l'ESP8266.
- **ESP.getFreeSketchSpace()** : Indique l'espace mémoire disponible restant pour téléverser un sketch supplémentaire.