

Programmation de l'EEPROM



Objectifs

- L'EEPROM d'Arduino Uno
- Accéder à l'EEPROM
- Exemple
 - Lecture
 - Écriture

EEPROM Arduino Uno

- EEPROM : Electrically Erasable Programmable Read-Only Memory
- Capacité : 1024 octets
- Limite de cycles d'écritures : 100 000 par octet (byte)
 - Limiter le nombre d'écritures
 - Limiter les écritures dans les mêmes zones : utiliser des zones différentes de la mémoire
- Les données peuvent être corrompues autour de cette valeur
 - Utilisation du contrôle de redondance cyclique - CRC (Cyclic Redundancy Check)
- Assimilable à un tableau de 1024 octets, c'est à dire un : « byte mémoire[1024] »

Accès à l'EEPROM – Bibliothèque

- Inclure les déclarations de la bibliothèque EEPROM :
`#include <EEPROM.h>`
- Documentation : <https://www.arduino.cc/en/Reference/EEPROM>
- Donne accès à un objet accessible à partir de la variable static EEPROM
- Capacité : `uint16_t length()`

Accès à l'EEPROM – Bibliothèque

- Méthodes bas niveau : manipulations d'un octet (byte)
 - `byte read(int adresse)`
 - `void write(int adresse, byte valeur) / void update(int adresse, byte valeur)`
 - `byte &operateur[int adresse]` : read / write
- Méthodes haut niveau : manipulation d'une type de base ou d'une structure / classe
 - `TypeValeur & get(int adresse, TypeValeur &valeur)`
 - `void put(int adresse, const TypeValeur &valeur)` : Utilise la méthode update

Accès à l'EEPROM – Bas niveau – Écriture

```
void effacementEEPROMAvecIndex() {  
    for (int i = 0 ; i < EEPROM.length() ; i++) {  
        EEPROM.update(i, 0);  
    }  
}
```

```
void remplirEEPROMAvecIndex() {  
    for (int i = 0 ; i < EEPROM.length() ; i++) {  
        EEPROM.update(i, i % 256);  
    }  
}
```

Accès à l'EEPROM – Bas niveau – Lecture

```
remplirEEPROMAvecIndex();  
afficherContenuEEPROMSurConsole();
```

```
void afficherContenuEEPROMSurConsole() {  
    Serial.println();  
  
    Serial.println("Capacite EEPROM : " + int2String(EEPROM.length(), DEC));  
  
    Serial.print(padLeft("", 4));  
    for (int compt = 0; compt < nbBytesParLigne ; ++compt) {  
        Serial.print(" " + padLeft(int2String(compt, HEX), 2, '0'));  
    }  
  
    for (int adresse = 0; adresse < EEPROM.length(); ++adresse) {  
        if (adresse % nbBytesParLigne == 0) {  
            Serial.println();  
            // String(adresse) ne fonctionne pas ici pour valeur > 100 !  
            Serial.print(padLeft(int2String(adresse, HEX), 4, '0'));  
        }  
  
        Serial.print(" " + padLeft(int2String( EEPROM.read(adresse), HEX), 2, '0'));  
    }  
}
```

Capacite EEPROM : 1024

0000	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
0010	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F
0020	20	21	22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F
0030	30	31	32	33	34	35	36	37	38	39	3A	3B	3C	3D	3E	3F
0040	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F
0050	50	51	52	53	54	55	56	57	58	59	5A	5B	5C	5D	5E	5F
0060	60	61	62	63	64	65	66	67	68	69	6A	6B	6C	6D	6E	6F
0070	70	71	72	73	74	75	76	77	78	79	7A	7B	7C	7D	7E	7F
0080	80	81	82	83	84	85	86	87	88	89	8A	8B	8C	8D	8E	8F
0090	90	91	92	93	94	95	96	97	98	99	9A	9B	9C	9D	9E	9F
00A0	A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	AA	AB	AC	AD	AE	AF
[...]																
03B0	B0	B1	B2	B3	B4	B5	B6	B7	B8	B9	BA	BB	BC	BD	BE	BF
03C0	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF
03D0	D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	DA	DB	DC	DD	DE	DF
03E0	E0	E1	E2	E3	E4	E5	E6	E7	E8	E9	EA	EB	EC	ED	EE	EF
03F0	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	FA	FB	FC	FD	FE	FF

Accès à l'EEPROM – Haut niveau – Lecture / Écriture

```
DonneesASauvegarder donneesAEcrire;
donneesAEcrire.nblignes = 12;      // 00 0C
donneesAEcrire.nbColonnes = 1104; // 04 50

ecrireDonnees(donneesAEcrire);
afficherContenuEEPROMSurConsole();

DonneesASauvegarder donneesLues;
lireDonnees(donneesLues);
Serial.println("nblignes: " + int2String(donneesLues.nblignes));
Serial.println("nbColonnes: " + int2String(donneesLues.nbColonnes));

void ecrireDonnees(const DonneesASauvegarder &p_das) {
    EEPROM.put(0x2a, p_das);
}

void lireDonnees(DonneesASauvegarder &p_das) {
    EEPROM.get(0x2a, p_das);
}
```

```
class DonneesASauvegarder {
public:
    int nblignes;
    int nbColonnes;
};
```


Accès à l'EEPROM – Haut niveau – Lecture / Écriture

```
DonneesASauvegarder donneesAEcrire;
donneesAEcrire.nbLignes = 12; // 00 0C
donneesAEcrire.nbColonnes = 1104; // 04 50

ecrireDonnees(donneesAEcrire);
afficherContenuEEPROMSurConsole();

DonneesASauvegarder donneesLues;
lireDonnees(donneesLues);
Serial.println("nbLignes: " + int2String(donneesLues.nbLignes));
Serial.println("nbColonnes: " + int2String(donneesLues.nbColonnes));

void ecrireDonnees(const DonneesASauvegarder &p_das) {
    EEPROM.put(0x2a, p_das);
}

void lireDonnees(DonneesASauvegarder &p_das) {
    EEPROM.get(0x2a, p_das);
}
```

```
class DonneesASauvegarder {
public:
    int nbLignes;
    int nbColonnes;
};
```

```
Capacite EEPROM : 1024
```

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
0000	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0010	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0020	00	00	00	00	00	00	00	00	00	00	0C	00	50	04	00	00
0030	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0040	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
[...]																
03F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

nbLignes: 12
nbColonnes: 1104

Accès à l'EEPROM – Haut niveau – Lecture / Écriture

```
DonneesASauvegarder donneesAEcrire;
donneesAEcrire.nblignes = 12;      // 00 0C
donneesAEcrire.nbColonnes = 1104; // 04 50

ecrireDonnees(donneesAEcrire);
afficherContenuEEPROMSurConsole();

DonneesASauvegarder donneesLues;
lireDonnees(donneesLues);
Serial.println("nblignes: " + int2String(donneesLues.nblignes));
Serial.println("nbColonnes: " + int2String(donneesLues.nbColonnes));

void ecrireDonnees(const DonneesASauvegarder &p_das) {
    EEPROM.put(0x2a, p_das);
}

void lireDonnees(DonneesASauvegarder &p_das) {
    EEPROM.get(0x2a, p_das);
}
```

```
class DonneesASauvegarder {
public:
    int nblignes;
    int nbColonnes;
};
```

Capacite EEPROM : 1024

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
0000	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0010	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0020	00	00	00	00	00	00	00	00	00	00	0C	00	50	04	00	00
0030	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
0040	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
[...]																
03F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

nblignes: 12
nbColonnes: 1104

=> Little endian

*Plus d'informations : <https://en.wikipedia.org/wiki/Endianness>

Accès à l'EEPROM – Limiter les écriture

- Toujours utiliser `update` ou `put`
- Se déplacer dans les adresses de l'EEPROM au bout de 50k écritures afin d'utiliser d'autres cases mémoire
 - Ex. garder l'adresse de début dans les deux premières cases, l'adresse du nombre d'écritures dans les deux suivantes
 - Vous pouvez aussi économiser de l'espace et prendre seulement les 3 premiers octets

Références

- <https://www.arduino.cc/en/Reference/EEPROM> : documentation Arduino
- <https://en.wikipedia.org/wiki/Endianness> : little endian / big endian
- <https://www.tinkercad.com/things/6YlRdo6KbQb> : démo du cours