

Systemes embarqués

# Mémoires permanentes pour les microcontrôleurs

Pierre-Yves Rochat

# Mémoires permanentes pour les microcontrôleurs

---

**Pierre-Yves Rochat**

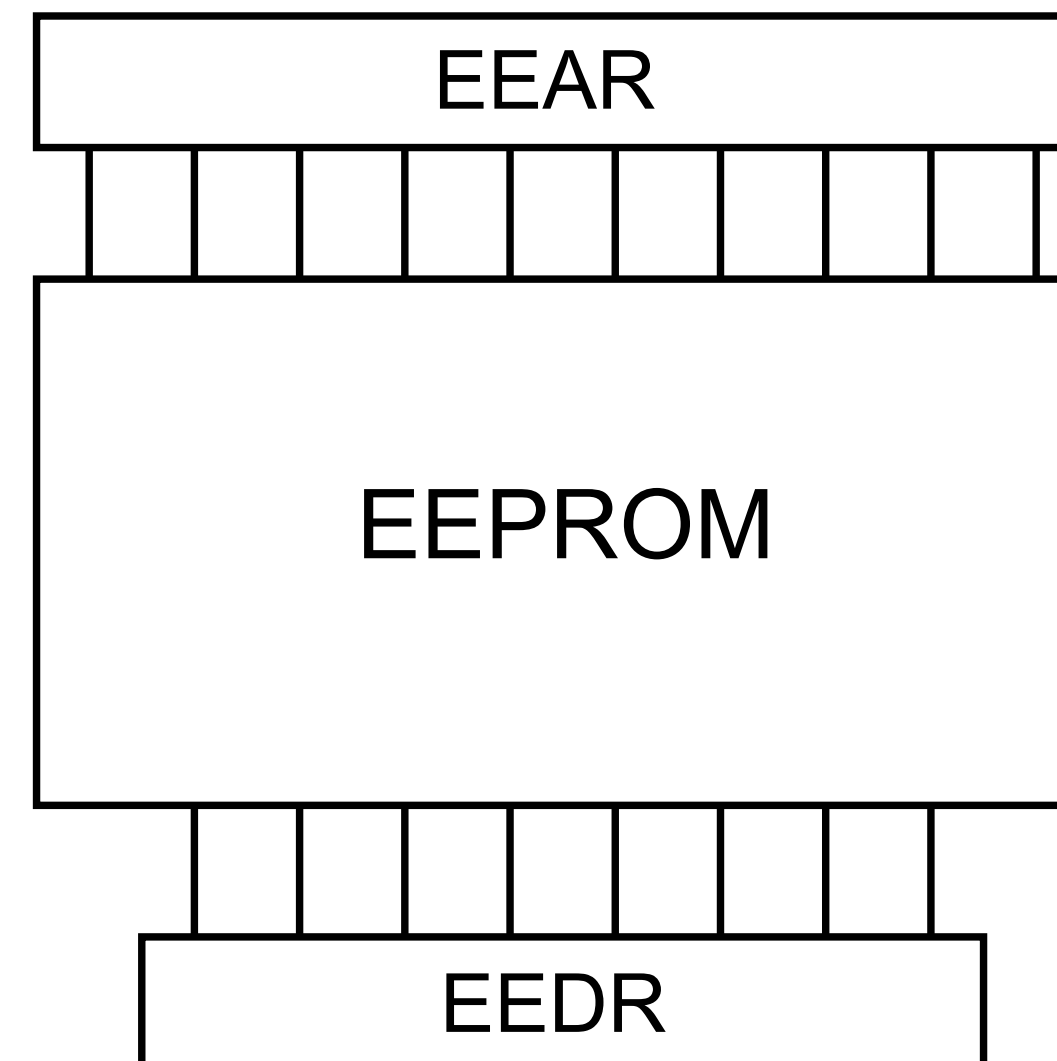
# Types de mémoires

---

- Principalement deux mémoires dans un microcontrôleur :
  1. Mémoire Flash, pour les programmes
  2. Mémoire vive (RAM) pour les données
- Comment conserver des données ?

Par exemple des paramètres de fonctionnement du système, des compteurs d'utilisation, etc.

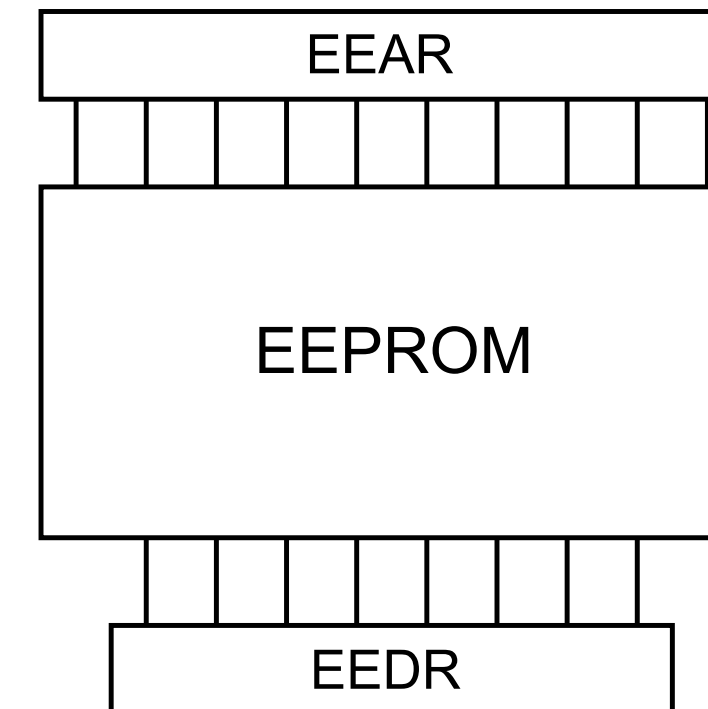
- Certains microcontrôleurs disposent d'une mémoire du type EEPROM  
*Electrically-Erasable Programmable Read-Only Memory*
- Par exemple, l'ATmega328 dispose de 1 kB d'EEPROM



# EEPROM : programmation

```

1 // Lecture en EEPROM :
2 EEAR = adresse; // l'adresse est donnée
3 EECR = (1<<EERE); // le fanion de lecture est activé
4 valeur = EEDR; // lecture de la valeur
5
6 // Ecriture en EEPROM :
7 while (EECR & (1<<EEPE)) {} // attente de la fin d'une éventuelle écriture précédente
8 EEAR = adresse; // l'adresse est donnée
9 EEDR = valeur; // la valeur est donnée
10 EECR = (1<<EEMPE); // autorise une écriture (Master Write Enable)
11 EECR = (1<<EEPE); // lance le cycle d'écriture (Write Enable)
  
```



- Il est généralement possible d'accéder à la mémoire flash d'un microcontrôleur ... en faisant attention à ne pas détruire le programme !
- C'est un peu plus difficile sur des microcontrôleurs dont l'architecture n'est pas du type Von Neumann
- Dans tous les cas, l'effacement s'effectue par bloc

# FLASH : programmation

```

1 // Lecture en Flash :
2 uint8_t *pointeur; // pointeur dans la Flash
3 pointeur = (uint8_t *) 0x1040; //place l'adresse dans le pointeur
4 uint8_t valeur = *pointeur;
5
6 // Ecriture en Flash :
7 FCTL3 = FWKEY; // Clear Lock bit
8 *pointeur = valeur; // écrit la valeur dans la Flash
9 FCTL3 = FWKEY + LOCK; // Set LOCK bit
10
11 // Effacement d'un bloc de la mémoire Flash
12 FCTL1 = FWKEY + ERASE; // Set Erase bit
13 FCTL3 = FWKEY; // Clear Lock bit
14 *pointeur = 0; // lance un cycle d'effacement du bloc, la valeur donnée n'a pas d'importance
15 FCTL3 = FWKEY + LOCK; // Set LOCK bit
16 FCTL1 = FWKEY; // Clear WRT bit
  
```

# Limite du nombre de cycles d'écriture

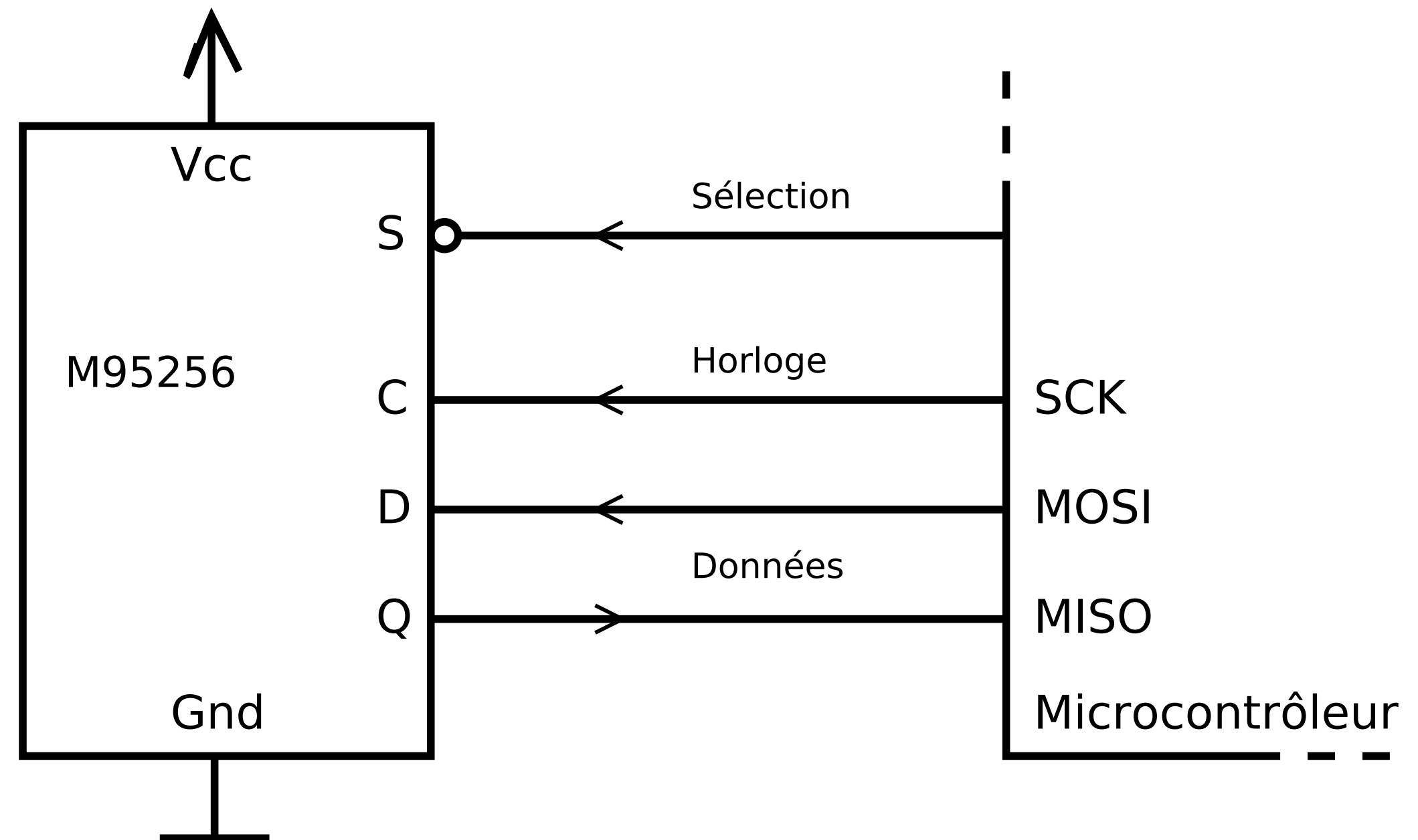
---

- Pour chaque type de mémoire non-volatile, un nombre de cycles limité
- Typiquement 10'000 pour une FLASH
- 100'000 pour une EEPROM



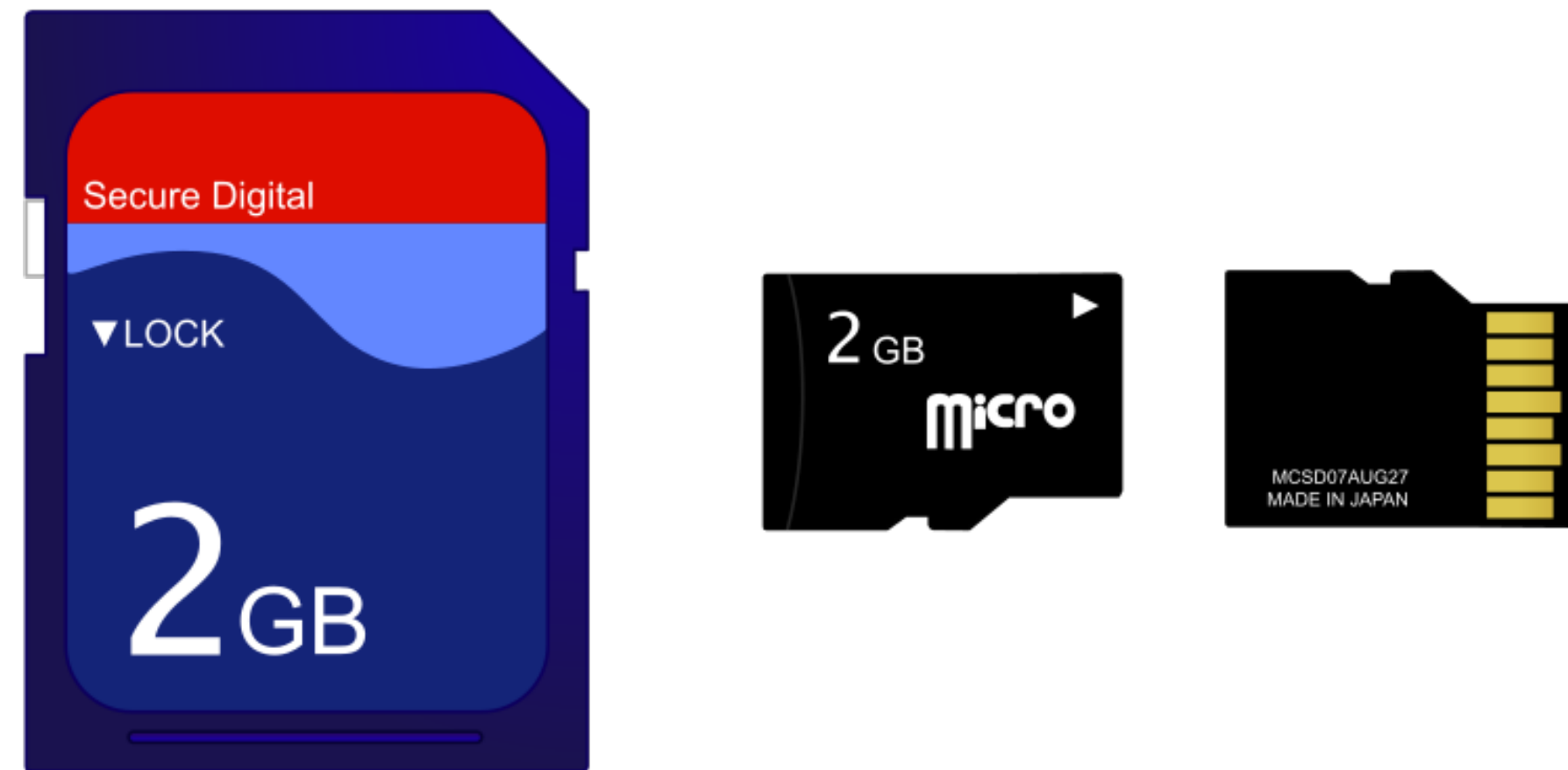
# Mémoires externes

- Il est possible d'ajouter des mémoires non-volatiles externes :
- RAM secourues
- EEPROM série (I2C ou SPI)



# Mémoires externes

- Il est possible d'ajouter des mémoires non-volatiles externes :
- RAM secourues
- EEPROM série
- Cartes SD



# Systeme de fichier

- Complicé de gérer les données d'une carte SD
- Pratique d'utiliser un système de fichier : par exemple FAT 32
- Des librairies sont disponibles

PetitFat :	<i>Procédure</i>	<i>Rôle</i>
	pf_mount:	Monter un volume
	pf_open:	Ouvrir un fichier
	pf_read:	Lire des données dans un fichier
	pf_write:	Écrire des données dans un fichier
	pf_lseek:	Déplacer le pointeur de lecture ou d'écriture
	pf_opendir:	Ouvrir un dossier
	pf_readdir:	Lire le contenu d'un dossier

# Mémoires permanentes pour les microcontrôleurs

---

- RAM : non-volatile
- EEPROM
- Mémoires Flash
- Mémoires externes

