

# Programmation cartes à puces

Loïc Demange

`loic.demange@etud.univ-paris8.fr`

avec les notes de cours de Philippe Guillot

31/4 février 2021



Pour compiler, programmer, et tester la carte.

Dans le terminal, dans le dossier du projet

- **make** pour compiler
- **make progcarte** pour programmer (la carte dans le programmeur)
- **scat hello.script** pour tester (la carte dans le lecteur)

Tester la carte avec SCAT, et rajouter des lignes dans `hello.script` pour tester la fonction `sortir_data()`.

L'EEPROM est une mémoire non volatile inscriptible par le programme.

Pour pouvoir l'utiliser, il faut

- inclure la bibliothèque avec **#include<avr/eeprom.h>**.
- déclarer les variables concernées avec le tag **EEMEM**.
- utiliser les différentes fonctions permettant d'écrire et de lire la mémoire EEPROM.

**Remarque** On nomme les variables avec le préfixe **ee\_** pour simplifier la lecture.

```
uint8_t ee_val EEMEM = 0;
```

```
// Permet de lire un entier de 8 octets dans l'EEPROM
uint8_t eeprom_read_byte(uint8_t * dest);

// Permet de lire un mot (entier de 16 octets) dans l'EEPROM
uint16_t eeprom_read_word(uint16_t * dest);

// Permet de lire n octets de données dans l'EEPROM
void eeprom_read_block(void * dest, void * src, uint8_t n);

// Permet d'ecrire un entier de 8 octets dans l'EEPROM
void eeprom_write_byte(uint8_t * dest, uint8_t val);

// Permet d'ecrire un mot (entier de 16 octets) dans l'EEPROM
void eeprom_write_word(uint16_t * dest, uint16_t val);

// Permet d'ecrire n octets de données dans l'EEPROM
void eeprom_write_block(void * src, void * dest, uint8_t n);
```

# Projet porte monnaie

Le but du projet est de réaliser un porte monnaie électronique, avec pour fonctionnalités

- introduction des données personnelles
- lecture des données personnelles
- lecture du solde
- dépôt d'argent
- retrait d'argent

Le solde sera sur 16 bits, et stocké dans un entier non signé (**uint16\_t**). Les données personnelles et le solde seront évidemment dans une mémoire non volatile.

La classe de la carte sera la classe libre **0x81**.

Deux problématiques à prendre en compte

- l'Endianness
- les dépôts ou retraits trop importants