

# La Carte à Microprocesseur

## Un système embarqué en plein essor

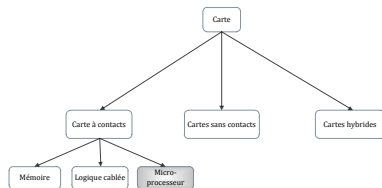
**Dr. Tegawendé F. Bissyandé**  
tegawende.bissyande@fasolabs.org

*Cours préparé pour*  
L'Institut Supérieur de Technologie (IST Burkina)

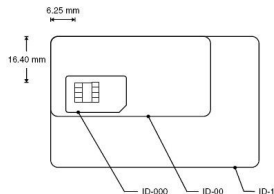
29 Février 2015

# Rappel chapitres 1 & 2

## Familles de cartes à puces



## Personnalisable!

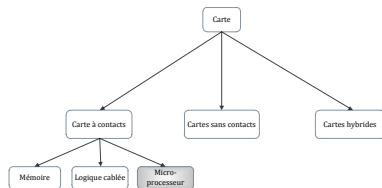


## En bref...

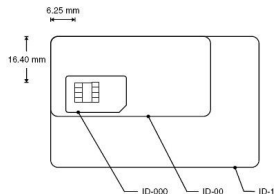
- 1 Différents domaines d'applications (banque, telecom, badges acces, ...)
- 2 Normalisation : caractéristiques physiques (tailles et résistances + niveaux électriques)
- 3 Brochage de la carte à puce : Pattes d'entrée des signaux (e.g., I/O)
- 4 Memoire de la carte à puce (ROM, RAM, EEPROM)
- 5 Microprocesseur : l'intelligence embarquée

# Rappel chapitres 1 & 2

## Familles de cartes à puces



## Personnalisable!

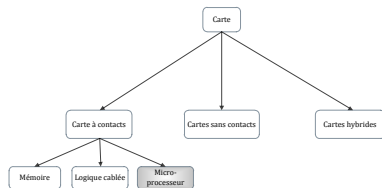


## En bref...

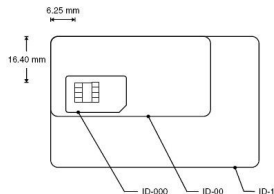
- 1 Différents domaines d'applications (banque, telecom, badges acces, ...)
- 2 Normalisation : caractéristiques physiques (tailles et résistances + niveaux électriques)
- 3 Brochage de la carte à puce : Pattes d'entrée des signaux (e.g., I/O)
- 4 Memoire de la carte à puce (ROM, RAM, EEPROM)
- 5 Microprocesseur : l'intelligence embarquée

# Rappel chapitres 1 & 2

## Familles de cartes à puces



## Personnalisable!

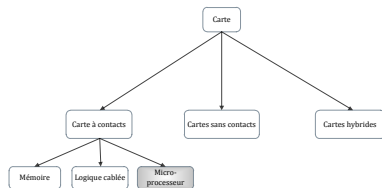


## En bref...

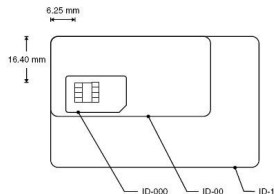
- 1 Différents domaines d'applications (banque, telecom, badges acces, ...)
- 2 Normalisation : caractéristiques physiques (tailles et résistances + niveaux électriques)
- 3 Brochage de la carte à puce : Pattes d'entrée des signaux (e.g., I/O)
- 4 Memoire de la carte à puce (ROM, RAM, EEPROM)
- 5 Microprocesseur : l'intelligence embarquée

# Rappel chapitres 1 & 2

## Familles de cartes à puces



## Personnalisable!

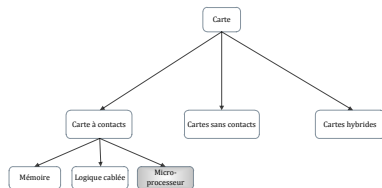


## En bref...

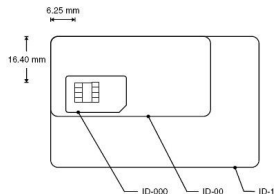
- 1 Différents domaines d'applications (banque, telecom, badges acces, ...)
- 2 Normalisation : caractéristiques physiques (tailles et résistances + niveaux électriques)
- 3 Brochage de la carte à puce : Pattes d'entrée des signaux (e.g., I/O)
- 4 Memoire de la carte à puce (ROM, RAM, EEPROM)
- 5 Microprocesseur : l'intelligence embarquée

# Rappel chapitres 1 & 2

## Familles de cartes à puces



## Personnalisable!

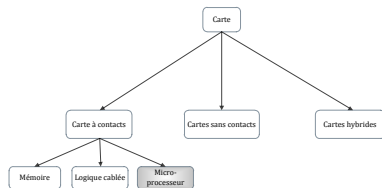


## En bref...

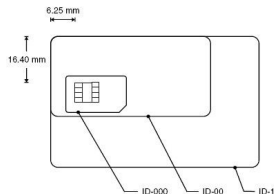
- 1 Différents domaines d'applications (banque, telecom, badges acces, ...)
- 2 Normalisation : caractéristiques physiques (tailles et résistances + niveaux électriques)
- 3 Brochage de la carte à puce : Pattes d'entrée des signaux (e.g., I/O)
- 4 Memoire de la carte à puce (ROM, RAM, EEPROM)
- 5 Microprocesseur : l'intelligence embarquée

# Rappel chapitres 1 & 2

## Familles de cartes à puces



## Personnalisable!



## En bref...

- 1 Différents domaines d'applications (banque, telecom, badges acces, ...)
- 2 Normalisation : caractéristiques physiques (tailles et résistances + niveaux électriques)
- 3 Brochage de la carte à puce : Pattes d'entrée des signaux (e.g., I/O)
- 4 Memoire de la carte à puce (ROM, RAM, EEPROM)
- 5 Microprocesseur : l'intelligence embarquée

# Modèle de communication & de stockage

## Communication avec le monde extérieur $\Rightarrow$ Interopérabilité

- Initialisation de la communication (Introduction dans le lecteur)
- Caractéristiques électriques (cf. Chapitre 1)
- Transmission de données
- Protocoles de transmissions
  - 1 Protocole T=0 (octet par octet)
  - 2 Protocole T=1 (block par block)
  - 3 Protocole PTS (négociation du protocol)

## Système de fichiers

- Organisation logique
- Structures d'un Fichier Elementaire
- Structure d'un message



# Modèle de communication & de stockage

## Communication avec le monde extérieur $\Rightarrow$ Interopérabilité

- Initialisation de la communication (Introduction dans le lecteur)
- Caractéristiques électriques (cf. Chapitre 1)
- Transmission de données
- Protocoles de transmissions
  - 1 Protocole T=0 (octet par octet)
  - 2 Protocole T=1 (block par block)
  - 3 Protocole PTS (négociation du protocol)

## Système de fichiers

- Organisation logique
- Structures d'un Fichier Elementaire
- Structure d'un message

# Modèle de communication & de stockage

## Communication avec le monde extérieur $\Rightarrow$ Interopérabilité

- Initialisation de la communication (Introduction dans le lecteur)
- Caractéristiques électriques (cf. Chapitre 1)
- Transmission de données
- Protocoles de transmissions
  - 1 Protocole T=0 (octet par octet)
  - 2 Protocole T=1 (block par block)
  - 3 Protocole PTS (négociation du protocol)

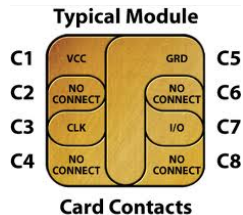
## Système de fichiers

- Organisation logique
- Structures d'un Fichier Elementaire
- Structure d'un message

# Caractéristiques électriques (Rappel)

## Garder le brochage en vue...

- VCC : Source de courant
- GND : Masse ou voltage de référence
- CLK : Horloge
- $V_{pp}$  : Voltage de programmation
- RST : Signal de reset
- I/O : Entrée/sortie série



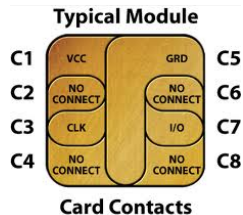
## À noter:

- 1 L'énergie de fonctionnement de la carte est fournie par le lecteur - VCC
- 2 La vitesse des échanges sur l'E/S est définie par la fréquence de l'horloge (externe) - CLK
- 3 Il faut une haute tension pour écrire sur la mémoire non volatile -  $V_{pp}$
- 4 Pour lancer le programme contenu dans la ROM, on utilise le signal de reset - RST
- 5 Une seule ligne pour l'échange des données entre la carte et l'interface - I/O

# Caractéristiques électriques (Rappel)

## Garder le brochage en vue...

- VCC : Source de courant
- GND : Masse ou voltage de référence
- CLK : Horloge
- $V_{pp}$  : Voltage de programmation
- RST : Signal de reset
- I/O : Entrée/sortie série



## À noter:

- 1 L'énergie de fonctionnement de la carte est fournie par le lecteur - VCC
- 2 La vitesse des échanges sur l'E/S est définie par la fréquence de l'horloge (externe) - CLK
- 3 Il faut une haute tension pour écrire sur la mémoire non volatile -  $V_{pp}$
- 4 Pour lancer le programme contenu dans la ROM, on utilise le signal de reset - RST
- 5 Une seule ligne pour l'échange des données entre la carte et l'interface - I/O

# Sequences pour (des)activation du Circuit intégré

## Activation

- Prendre RST au niveau bas
- Appliquer une tension  $V_{CC}$
- Mettre la ligne I/O en mode reception
- Mettre  $V_{pp}$  en mode veille
- Appliquer l'horloge CLK
- Prendre RST au niveau haut

## Désactivation

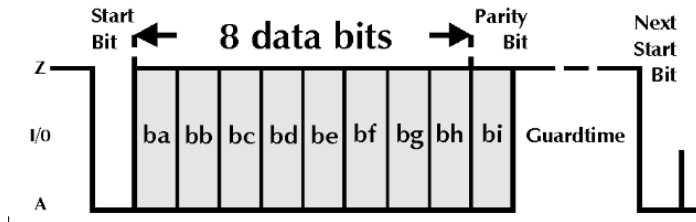
- Prendre RST au niveau bas
- Prendre l'horloge CLK au niveau bas
- Désactiver  $V_{pp}$
- Mettre la ligne I/O dans l'état bas
- Désactiver  $V_{CC}$



# Transmission d'un caractère

## caracteristiques :

- Transmission d'un octet (8 bits)
- Mode semi-duplex (à *chacun son tour de parler*, car une seule ligne I/O)
- Utilisation d'un **Start bit** pour synchroniser la trame
- Utilisation d'un **Parity bit** pour détecter les erreurs
- Utilisation d'un **Guardtime** pour la séparation entre les octets

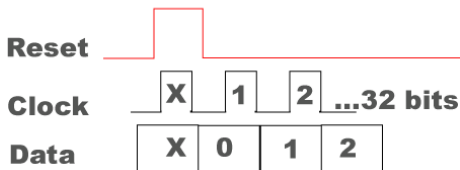


# Answer to reset

## Répondre à un signal d'enclenchement :

- Une réponse entre 400 et 40.000 cycles d'horloge (*apres le passage au niveau haut*)
- Une réponse en 33 octets au plus, et avec 5 champs :
  - 1 TS - **l'octet initial** : pattern de synchronisation (sens de logique/taux de transfert)
  - 2 T0 - **l'octet de format** : comment interpréter la réponse
  - 3  $TA_i, TB_i, TC_i, TD_i$  - **les octets d'interface** : informations de la carte (horloge, voltage de prog., protocol, etc...)
  - 4 T1, T2, TK - **les octets d'historique** : information sur le cycle de vie de la carte
  - 5 TCK - **l'octet de vérification** : Une espece de checksum pour vérifier l'intégrité

### Answer to reset



# Protocoles de transmission des données

## Protocoles

- 1 Peut-on changer de protocoles?
- 2 Modes *négociable* et *spécifique*
- 3  $\Rightarrow TA_2$  présent ou pas dans l'ATR (answer to reset)

## Proctocole $T = 0$

- Asynchrone, semi-duplex, et transmission **octet par octet**

## Protocole de sélection...

- 1 Protocol Type Selection (PTS)
- 2 Doit être envoyé immédiatement apres l'ATR
- 3 N'est applicable que si 'en mode negociable'.

## Proctocole $T = 1$

- Asynchrone, semi-duplex, et transmission **block par block**



# Une dernière chose... le système de fichier

## Qu'est-ce qui fait un OS?

- Un processeur : (e.g., microcontrôleur)
- De la mémoire pour des données de sauvegarde (e.g., ROM)
- De la mémoire pour des données de travail (e.g., RAM)
- De la mémoire pour des données d'applications (e.g. EEPROM)
- Des droits d'accès utilisateurs (e.g., sous UNIX  $\Rightarrow$  ACL)
- Un système de fichier

## Dans une carte à puce

- 1 MF : Master File (racine/; info de contrôle)
- 2 DF : Dedicated File (comme un répertoire)
- 3 EF : Elementary File (contient les données)



# Une dernière chose... le système de fichier

## Qu'est-ce qui fait un OS?

- Un processeur : (e.g., microcontrôleur)
- De la mémoire pour des données de sauvegarde (e.g., ROM)
- De la mémoire pour des données de travail (e.g., RAM)
- De la mémoire pour des données d'applications (e.g. EEPROM)
- Des droits d'accès utilisateurs (e.g., sous UNIX  $\Rightarrow$  ACL)
- Un système de fichier

## Dans une carte à puce

- 1 MF : Master File (racine/; info de contrôle)
- 2 DF : Dedicated File (comme un répertoire)
- 3 EF : Elementary File (contient les données)



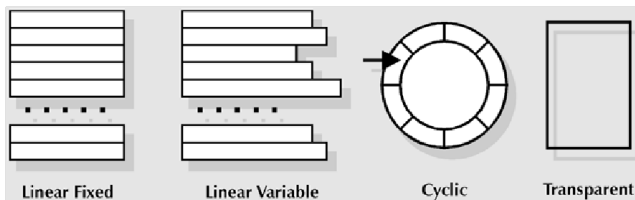
# Les fichiers élémentaires

## Types d'EF

- ① Working File : stocke les données d'app
- ② Public File : données accessibles sans condition
- ③ Application Control File : accès en lecture
- ④ Internal Secret File : Aucun accès depuis l'exterieur de la carte

## Dans une carte à puce

- ① Linear fixed
- ② Linear variable
- ③ Cyclic
- ④ Transparent (juste un bloc)



# Les fichiers élémentaires

## Types d'EF

- ① Working File : stocke les données d'app
- ② Public File : données accessibles sans condition
- ③ Application Control File : accès en lecture
- ④ Internal Secret File : Aucun accès depuis l'exterieur de la carte

## Dans une carte à puce

- ① Linear fixed
- ② Linear variable
- ③ Cyclic
- ④ Transparent (juste un bloc)

