

# Travaux pratiques informatique industrielle 1

Adimou Hardy  
Chebli Sonia  
De Castro Julie

2 avril 2025

## 1 Programmation d'un chenillard et application à la commande de moteur pas-à-pas

### 1.1 Programme CHENILLARD sur le port B

#### a) Schéma du chenillard

PTB7	PTB6	PTB5	PTB4	PTB3	PTB2	PTB1	PTB0
							X (\$01)
						X (\$02)	
					X (\$04)		
				X (\$08)			
			X (\$10)				
		X (\$20)					
	X (\$40)						
X (\$80)							

#### b) Voir programme

```
; variable/data section
;
;          ORG    $0200      ; table de données en Ram à partir de l'adresse $0200
;          FCB    0,1,2,4,8,16,32,64,128
;
; code section
;
;          ORG    ROMStart

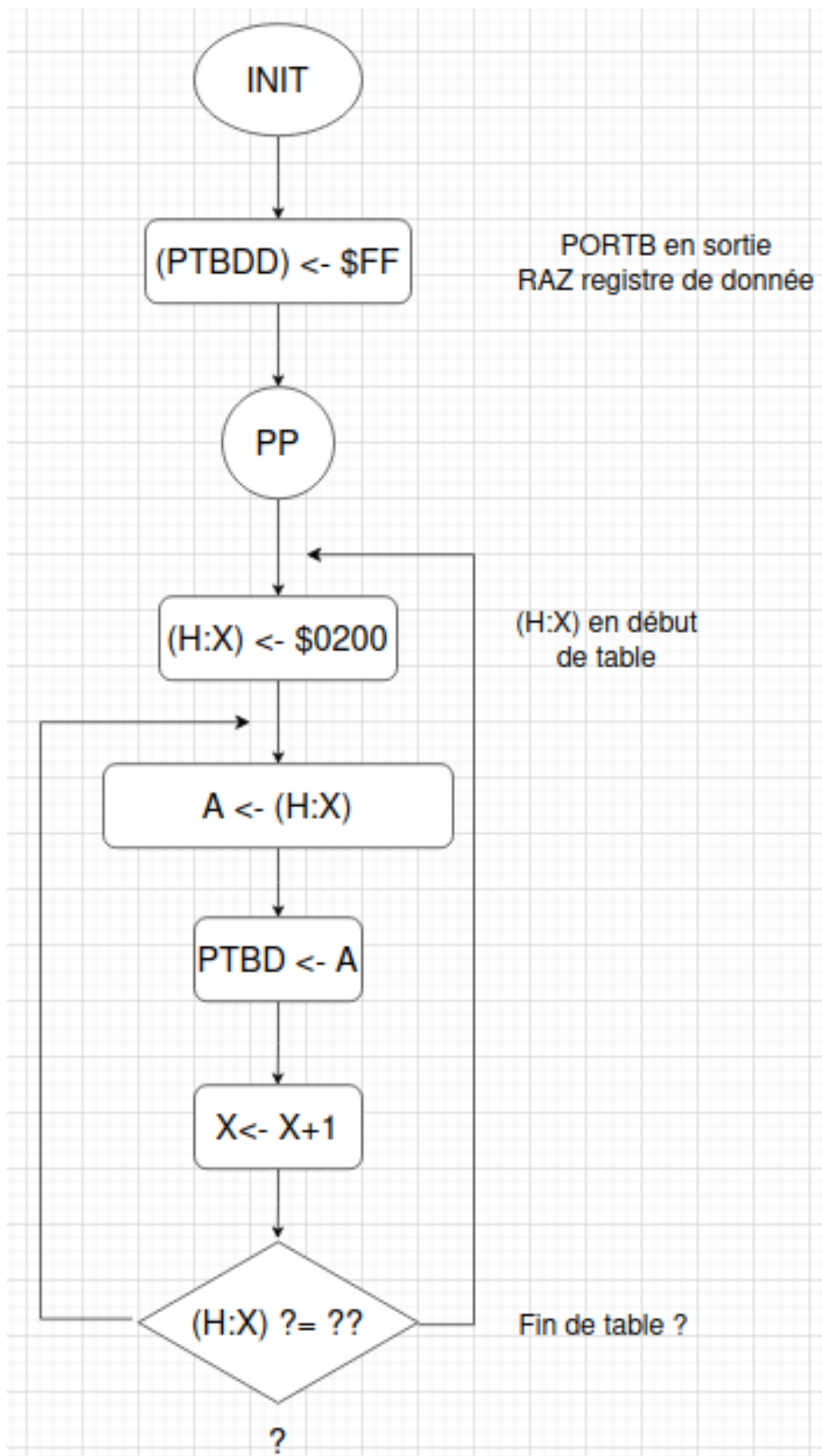
_Startup:
    LDHX    #RAMEnd+1        ; initialize the stack pointer
    TXS
    CLI                      ; enable interrupts
    LDA    #42
    STA    SOPT1
    MOV    #$FF, PTBDD

mainLoop:
    LDHX    #$0200
    LDA    ,X
    STA    PTBD
    STHX    $0250
    JSR    ATTENTE
    LDHX    $0250
    INCX
    CPHX    #$0209
    BNE    t
    BRA    mainLoop

;S/PROG
    ORG    $FB00

ATTENTE:
    LDHX    #$FFFF
    AIX    #-1
    CPHX    #0
    BNE    b
    RTS
```

c) Algorithme du programme principal



#### d) Voir programme

```

; variable/data section
;
; ORG    $0200    ; table de données en Ram à partir de l'adresse $0200
; FCB    0,1,2,4,8,16,32,64,128
;
; code section
;
; ORG    ROMStart

_Startup:
    LDHX    #RAMEnd+1    ; initialize the stack pointer
    TXS
    CLI        ; enable interrupts
    LDA    #42
    STA    SOPT1
    MOV    #$FF, PTBDD

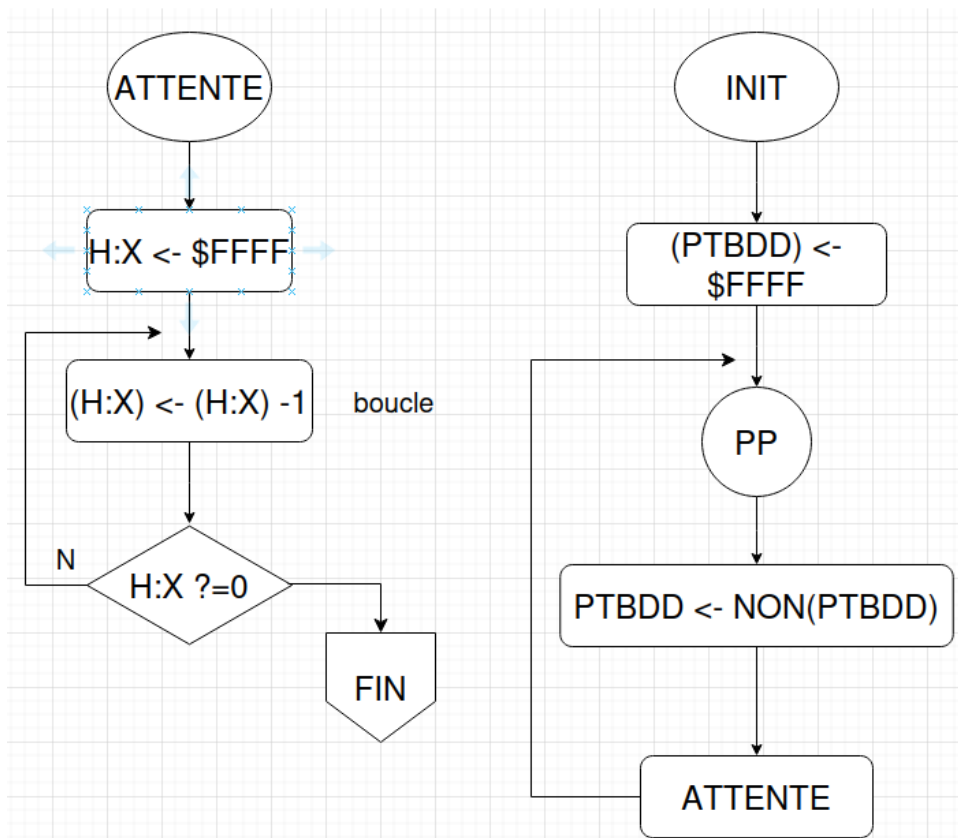
mainLoop:
    LDHX    #$0200
    LDA    ,X
    STA    PTBD
    STHX    $0250
    JSR    ATTENTE
    LDHX    $0250
    INCX
    CPHX    #$0209
    BNE    t
    BRA    mainLoop

;S/PROG
; ORG    $FB00

ATTENTE:
    LDHX    #$FFFF
    AIX    #-1
    CPHX    #0
    BNE    b
    RTS
    b

```

#### e) Algorithme du sous-programme ATTENTE



## f) Voir programme

```
; variable/data section
;
;      ORG      $0200      ; table de données en Ram à partir de l'adresse $0200
;      FCB      0,1,2,4,8,16,32,64,128
;
; code section
;
;      ORG      ROMStart

_Startup:
    LDHX      #RAMEnd+1      ; initialize the stack pointer
    TXS
    CLI                      ; enable interrupts
    LDA      #42
    STA      SOPT1
    MOV      #$FF, PTBDD

mainLoop:
    LDHX      #$0200
    LDA      ,X
    STA      PTBD
    STHX      $0250
    JSR      ATTENTE
    LDHX      $0250
    INCX
    CPHX      #$0209
    BNE      t
    |
    BRA      mainLoop

;S/PROG
;      ORG      $FB00

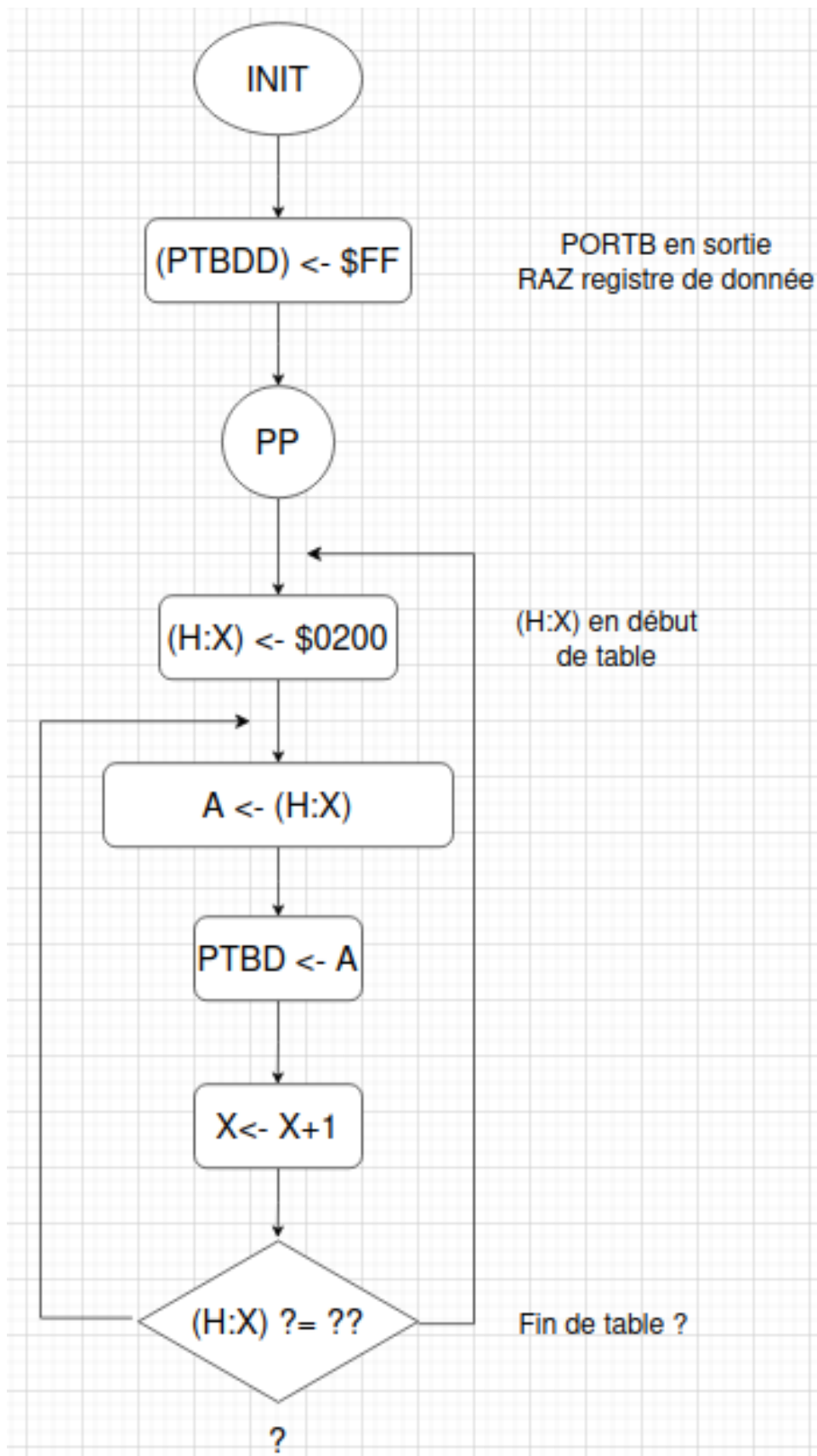
ATTENTE:
    LDHX      #$FFFF
    AIX      #-1
    CPHX      #0
    BNE      b
    RTS
    b
```

## 1.2 Commande par microcontrôleur d'un moteur pas-à-pas

### a)

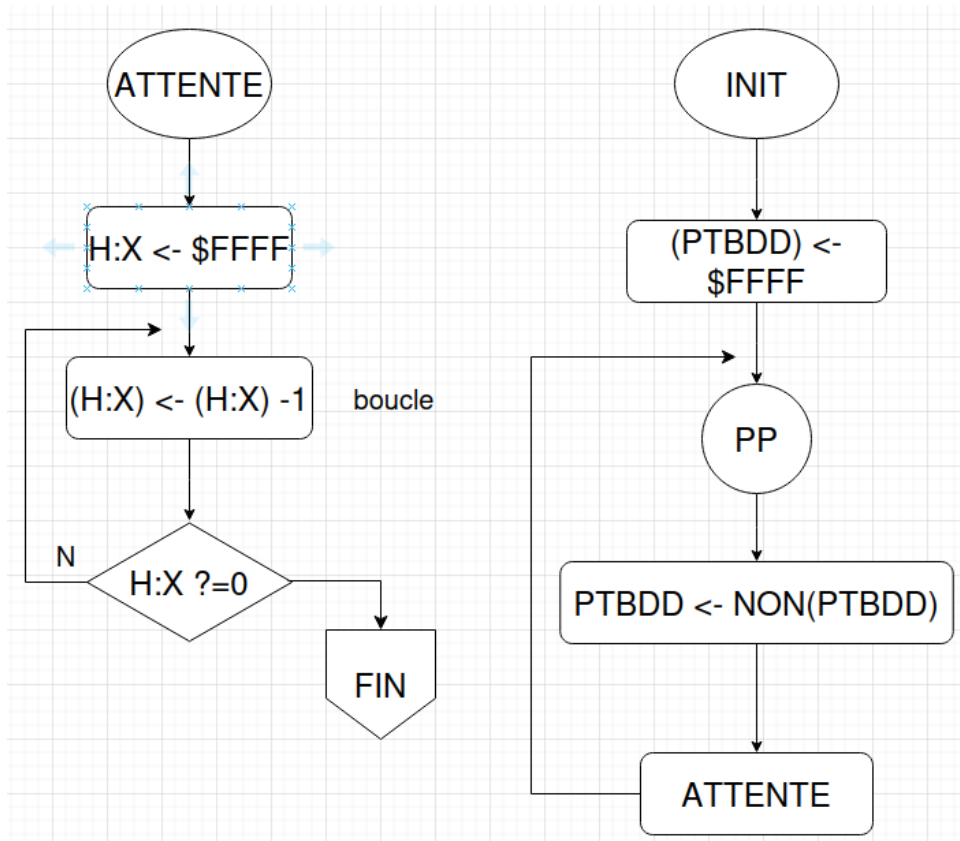
Avec la table :

- Beaucoup de lignes de code
- Tourne lentement



Sans la table :

- Moins de lignes de code
- Tourne plus rapidement



b)

$\frac{360}{1,8} = 200$  pas pour faire un tour.

1 tour -> 15

$\frac{1}{200} = 0,005 = 5$  ms

65 535 -> 131 ms

? -> 5 ms

$\frac{5 \cdot 65535}{131} = 2500$

La rotation est très rapide que les leds ne clignote pas.

c) Il faut inverser les valeurs de la table.

d) Voir programme

```
; variable/data section
;
;          ORG    $0200      ; table de données en Ram à partir de l'adresse $0200
;          FCB    12,9,3,6
;
;
; code section
;
;          ORG    ROMStart

_Startup:
    LDHX    #RAMEnd+1        ; initialize the stack pointer
    TXS
    CLI                      ; enable interrupts
    LDA    #42
    STA    SOPT1
    MOV    #$FF, PTBDD

mainLoop:
    LDHX    #$0200
t
    LDA    ,X
    STA    PTBD
    STHX    $0250
    JSR    ATTENTE
    LDHX    $0250
    INCX
    CPHX    #$0204
    BNE    t
    BRA     mainLoop

;S/PROG
;          ORG    $FB00

ATTENTE:
b
    LDHX    #$FFFF
    AIX    #-1
    CPHX    #0
    BNE    b
    RTS
```

### 1.3 Programmation du moteur pas-à-pas : DANS UN SENS PUIS DANS L'AUTRE

x25 parce que 4 pas  
 $\frac{200}{4} = 50$

```

                ORG     ROMStart

_Startup:       LDHX    #RAMEnd+1        ; initialize the stack pointer
                TXS
                CLI                    ; enable interrupts
                LDA     #42
                STA     SOPT1
                MOV     #$FF, PTBDD

mainLoop:       CLRA

;Sens anti horaire
sah             MOV     #12, PTBD
                JSR     ATTENTE
                MOV     #9, PTBD
                JSR     ATTENTE
                MOV     #3, PTBD
                JSR     ATTENTE
                MOV     #6, PTBD
                JSR     ATTENTE
                INCA
                CMP     #51
                BNE     sah

;Sens horaire
sh             MOV     #6, PTBD
                JSR     ATTENTE
                MOV     #3, PTBD
                JSR     ATTENTE
                MOV     #9, PTBD
                JSR     ATTENTE
                MOV     #12, PTBD
                JSR     ATTENTE
                DECA
                CMP     #0
                BNE     sh
                BRA     mainLoop

;S/PROG
                ORG     $FB00

ATTENTE:       LDHX    #2500
b              AIX     #-1
                CPHX    #0
                BNE     b
                RTS

```