

« Dauphin Web »

Développement d'un simulateur de
d'ordinateur 8 bits avec une interface web

Projet personnel - Formation GymInf

Nicolas Kunz (Gymnase d'Yverdon)

nicolas.kunz@eduvaud.ch

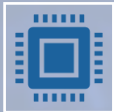
Superviseuse : Biljana Petreska von Ritter-Zahony





Introduction

Motivations
Simulateurs existants
Conception



Dauphin Web

Interface
Fonctionnement



Discussion

vs. simulateur de Dauphin
Validation
Améliorations



Activité



Retour sur l'activité

The background of the slide is a close-up, angled view of a circuit board. It features a complex network of glowing blue and orange lines representing circuit traces. In the upper right corner, there is a rectangular area displaying binary code (0s and 1s) in a glowing blue font. The overall color scheme is dark blue with vibrant highlights from the circuit traces.

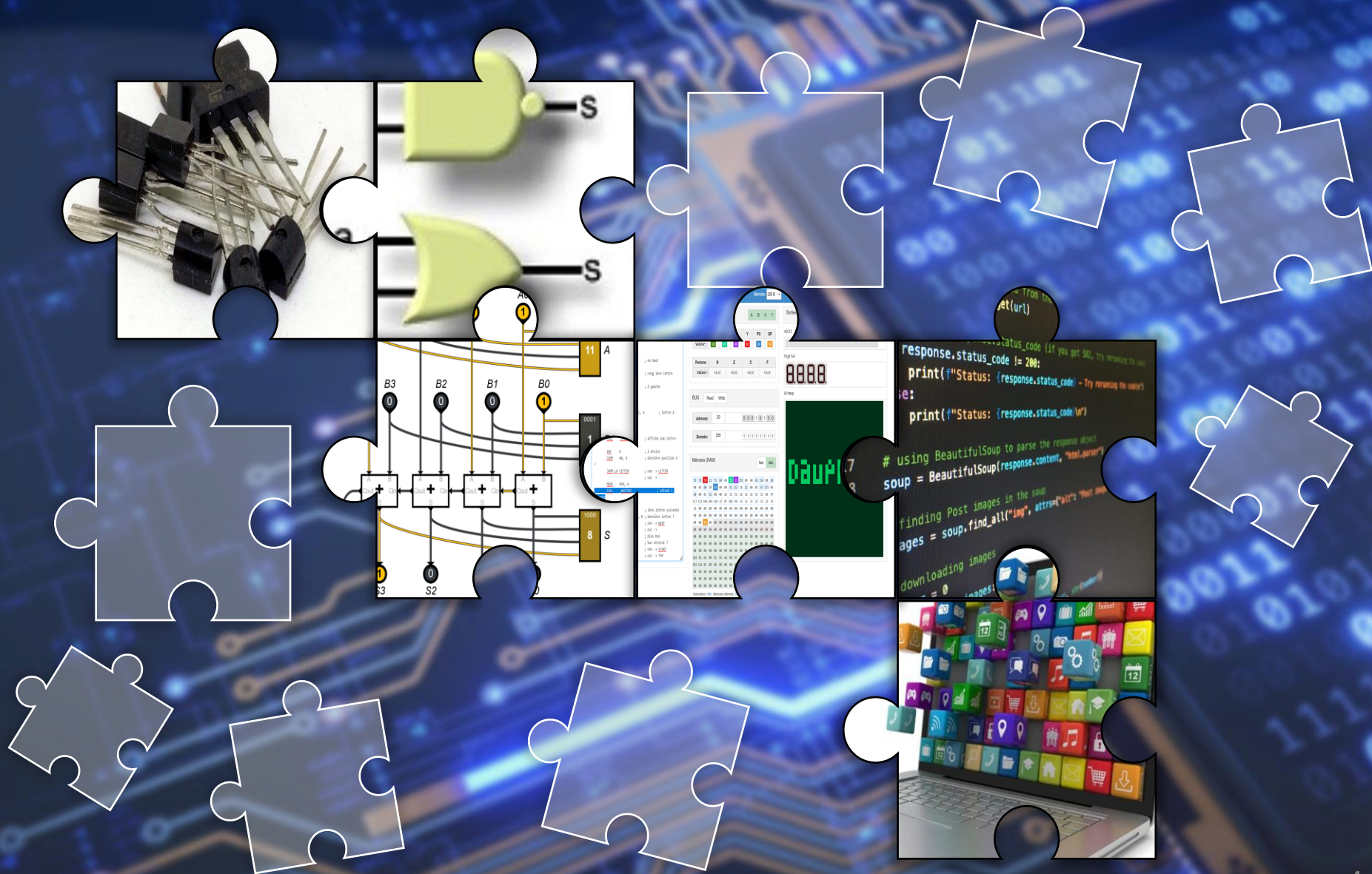
Introduction

Motivations

Simulateurs existants

Conception

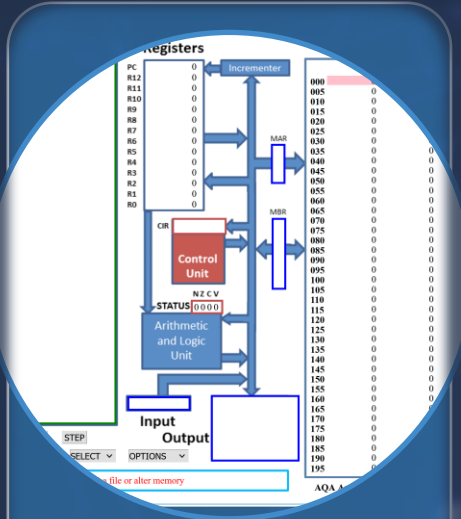
Motivations



Motivations



Simulateurs existants



AQA Processor Simulator

Peter Higginson

- + Flux des données
- Vite désorientant



ARMLite

Peter Higginson
+ Performance

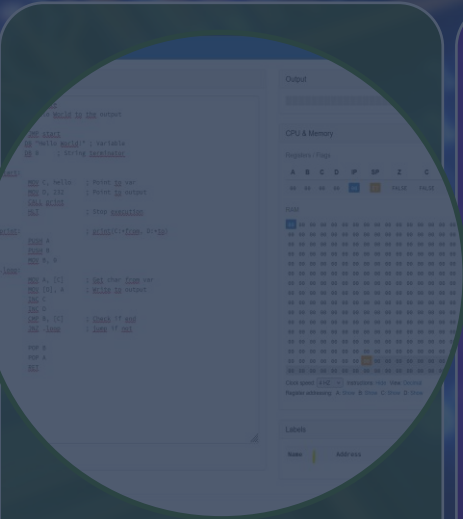
- Austère



Simple 8-bit Assembler Simulator

Marco Schweighauser

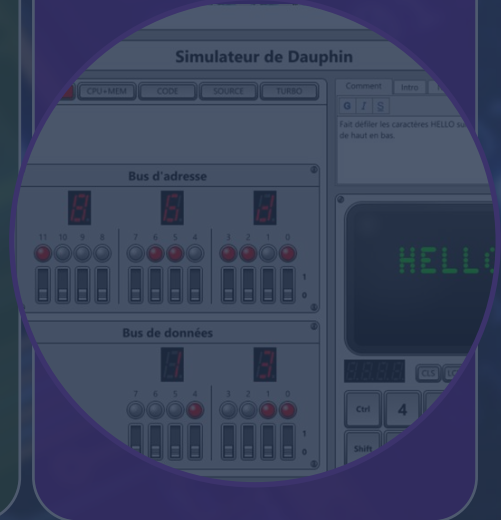
- + Interface utilisateur
- Limité



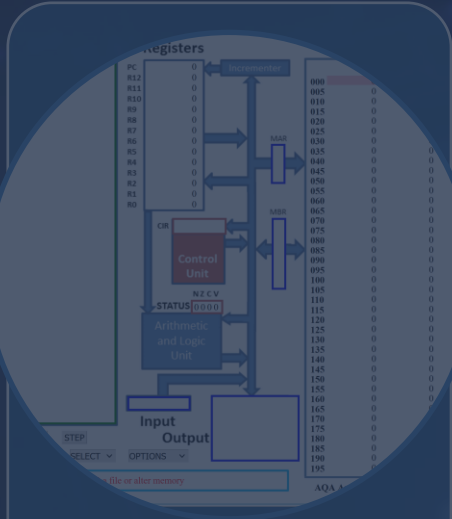
Simulateur de Dauphin

EPSITEC

- + Fonctionnalité
- Accessibilité



Simulateurs existants



AQA Processor Simulator

Peter Higginson

+ Flux des données
- Vite désorientant

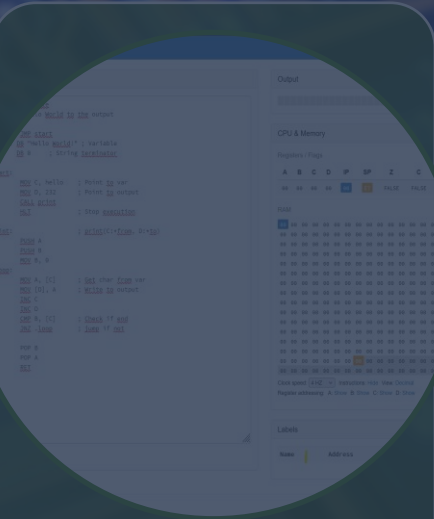
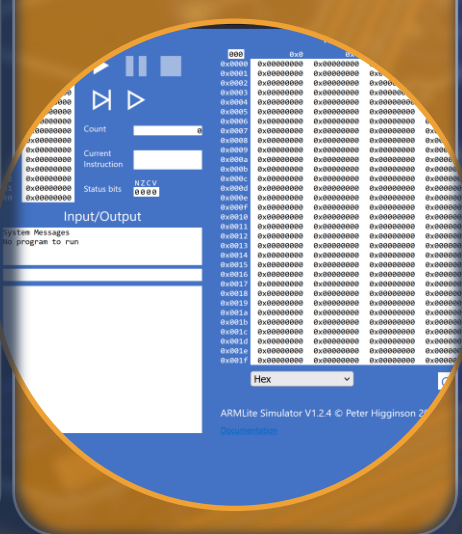


ARMLite

Peter Higginson

+ Performance

- Austère



Simple 8-bit Assembler Simulator

Marco Schweighauser

+ Interface utilisateur

- Limité



Simulateur de Dauphin

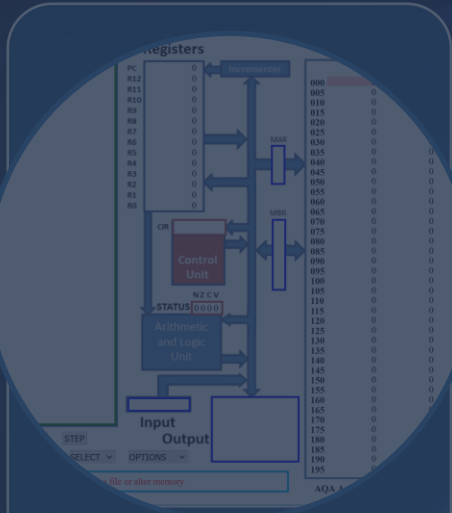
EPSITEC

+ Fonctionnalité

- Accessibilité



Simulateurs existants



AQA Processor Simulator

Peter Higginson

+ Flux des données
- Vite désorientant



ARMLite

Peter Higginson

+ Performance

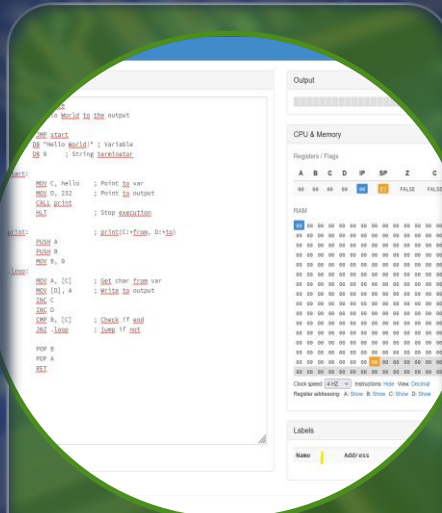
- Austère



Simple 8-bit Assembler Simulator

Marco Schweighauser

+ Interface utilisateur
- Limité



Simulateur de Dauphin

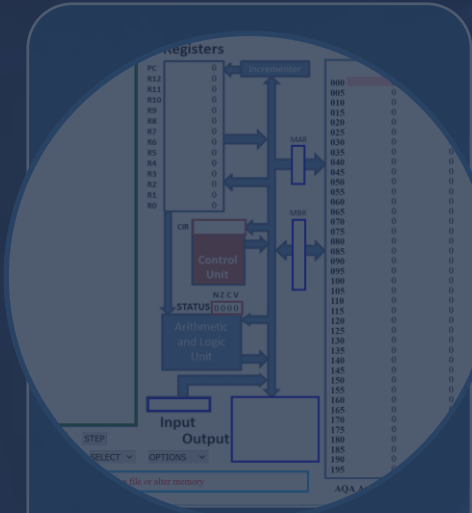
EPSITEC

+ Fonctionnalité

- Accessibilité



Simulateurs existants



AQA Processor Simulator

Peter Higginson

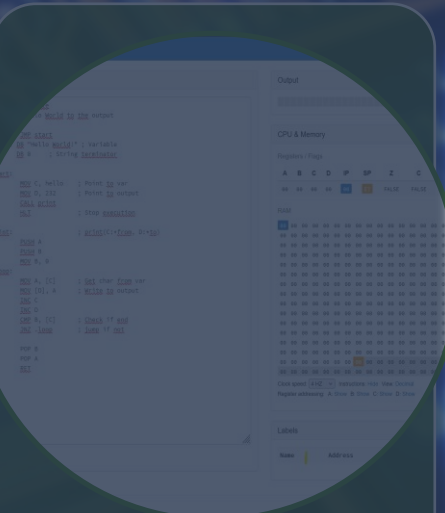
+ Flux des données
- Vite désorientant



ARMLite

Peter Higginson
+ Performance

- Austère



Simple 8-bit Assembler Simulator

Marco Schweighauser

+ Interface utilisateur
- Limité



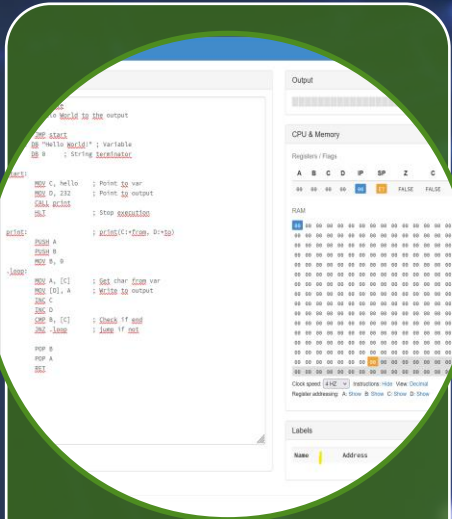
Simulateur de Dauphin

EPSITEC

+ Fonctionnalité
- Accessibilité



Conception

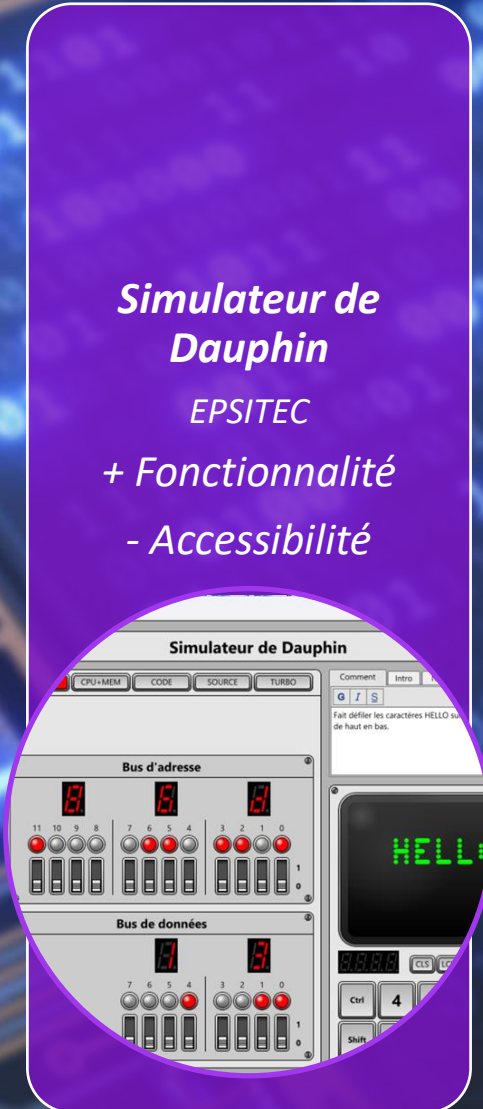


Simple 8-bit Assembler Simulator

Marco Schweighauser

***+ Interface
utilisateur***

- Limité



***Simulateur de
Dauphin***
EPSITEC
+ Fonctionnalité
- Accessibilité

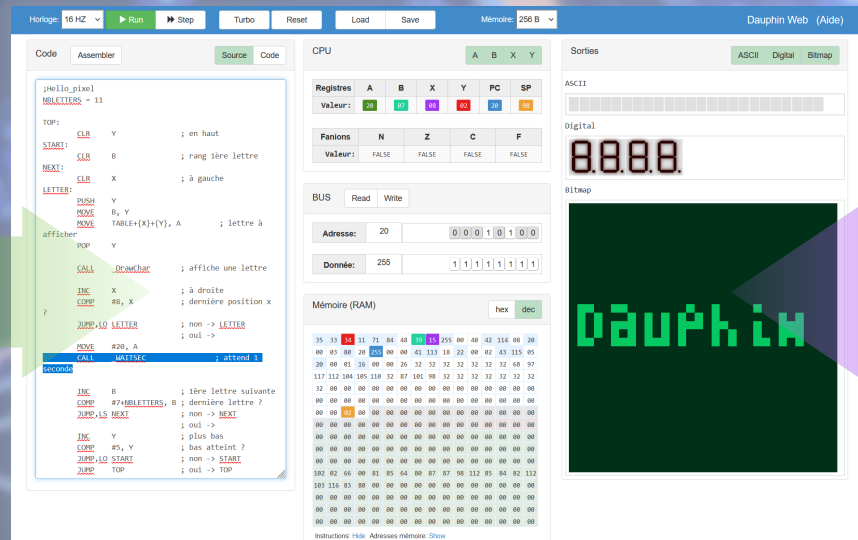
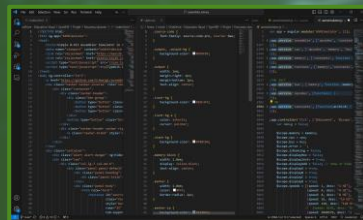
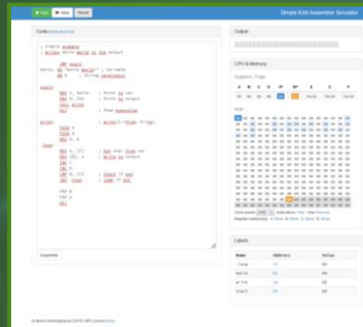
Simple 8-bit Assembler Simulator



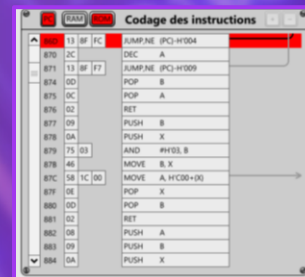
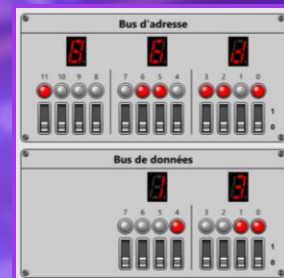
Interface

Code

Conception



Simulateur de Dauphin



BUS

I/O

Assembleur

Simple 8-bit
Assembler Simulator

Conception

Simulateur de

Dauphin Web (Aide)

Horloge: 16 HZ



Step

Turbo

Reset

Load

Save

Mémoire: 256 B

Code Assembler

Source Code

```
;Hello_pixel
NBLETTERS = 11

TOP:
START: CLR Y ; en haut
NEXT: CLR B ; rang 1ère lettre
LETTER: CLR X ; à gauche
        PUSH Y
        MOVE B, Y
        MOVE TABLE+{X}+{Y}, A ; lettre à
afficher POP Y
        CALL _DrawChar ; affiche une lettre
        INC X ; à droite
        COMP #8, X ; dernière position x
?
        JUMP,LO LETTER ; non -> LETTER
        ; oui ->
        MOVE #20, A
        CALL WAITSEC ; attend 1
seconde
        INC B ; 1ère lettre suivante
        COMP #7+NBLETTERS, B ; dernière lettre ?
        JUMP,LS NEXT ; non -> NEXT
        ; oui ->
        INC Y ; plus bas
        COMP #5, Y ; bas atteint ?
        JUMP,LO START ; non -> START
        JUMP TOP ; oui -> TOP
```

CPU

A B X Y

Registres	A	B	X	Y	PC	SP
Valeur:	20	07	08	02	20	98

Fanions	N	Z	C	F
Valeur:	FALSE	FALSE	FALSE	FALSE

BUS

Read Write

Adresse: 20 0 0 0 1 0 1 0 0

Donnée: 255 1 1 1 1 1 1 1 1

Mémoire (RAM)

hex dec

35	33	34	11	71	84	48	39	15	255	00	40	42	114	08	20
00	03	80	20	255	00	00	41	113	18	22	00	02	43	115	05
20	00	01	16	00	00	26	32	32	32	32	32	32	32	68	97
117	112	104	105	110	32	87	101	98	32	32	32	32	32	32	32
32	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	02	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
102	02	66	00	81	85	64	80	87	87	98	112	85	84	82	112
103	116	83	80	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Instructions: Hide Adresses mémoire: Show

Sorties

ASCII Digital Bitmap

ASCII



Digital



Bitmap



Simple 8-bit
Assembler Simulator





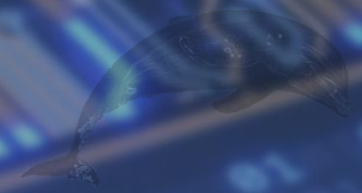
Dauphin Web

Interface

Fonctionnement

Instructions: [Hide](#) Adresses mémoire: [Show](#)

Interface



Horloge: 16 HZ ▶ Run ▶▶ Step Turbo Reset Load Save Mémoire: 256 B Dauphin Web (Aide)

Code

Assembler

Source

Code

```
;Hello_pixel
NBLETTERS = 11

TOP:

START:
  CLR    Y          ; en haut
  CLR    B          ; rang 1ère lettre
NEXT:
  CLR    X          ; à gauche
LETTER:
  PUSH   Y
  MOVE   R, Y

  INC    X          ; à droite
  COMP   #8, X      ; dernière position x
  ?
  JUMP,LO LETTER    ; non -> LETTER
                  ; oui ->
  MOVE   #20, A
  CALL   WAITSEC    ; attend 1
seconde

  INC    B          ; 1ère lettre suivante
  COMP   #7+NBLETTERS, B ; dernière lettre ?
  JUMP,LS NEXT      ; non -> NEXT
                  ; oui ->
  INC    Y          ; plus bas
  COMP   #5, Y      ; bas atteint ?
  JUMP,LO START     ; non -> START
  JUMP   TOP        ; oui -> TOP
```

CPU

A B X Y

Registres	A	B	X	Y	PC	SP
Valeur:	20	07	08	02	20	98

Fanions	N	Z	C	F
Valeur:	FALSE	FALSE	FALSE	FALSE

BUS Read Write

Sorties

ASCII

Digital

Bitmap

Horloge: 16 HZ ▶ Run ▶▶ Step Turbo Reset Load Save

Mémoire (RAM)

hex

dec

35	33	34	11	71	84	48	39	15	255	00	40	42	114	08	20
00	03	80	20	255	00	00	41	113	18	22	00	02	43	115	05
20	00	01	16	00	00	26	32	32	32	32	32	32	68	97	
117	112	104	105	110	32	87	101	98	32	32	32	32	32	32	32
32	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	02	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
102	02	66	00	81	85	64	80	87	87	98	112	85	84	82	112
103	116	83	80	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Instructions: [Hide](#) Adresses mémoire: [Show](#)

Interface



Horloge: 16 HZ ▾ ▶ Run ⏸ Step Turbo Reset Load Save Mémoire: 256 B ▾ Dauphin Web (Aide)

Code Assembler Source Code

```
;Hello_pixel
NBLETTERS = 11
```

CPU A B X Y PC SP

Registres	A	B	X	Y	PC	SP
Valeur:	20	07	08	02	20	98

Sorties ASCII Digital Bitmap

ASCII

Horloge: 16 HZ ▾ ▶ Run ⏸ Step Turbo Reset Load Save

Code Assembler

```
CLR X ; à gauche
LETTER:
PUSH Y
MOVE B, Y
MOVE TABLE+(X)+(Y), A ; lettre à
```

BUS Read Write

hex dec

0 0 0 1 0 1 0 0

1 1 1 1 1 1 1 1

55 00 40 42 114 08 20

18 22 00 02 43 115 05

32 32 32 32 32 68 97

32 32 32 32 32 32 32

00 00 00 00 00 00 00

00 00 00 00 00 00 00

00 00 00 00 00 00 00

00 00 00 00 00 00 00

00 00 00 00 00 00 00

00 00 00 00 00 00 00

102 02 66 00 81 85 64 80 87 87 98 112 85 84 82 112

103 116 83 80 00 00 00 00 00 00 00 00 00 00 00 00

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

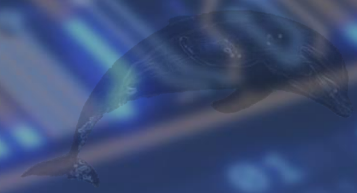
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

TOP: ; OUT TOP

Instructions: Hide Adresses mémoire: Show

Pause Stop

Interface



Horloge: 16 HZ

Run

Step

Turbo

Reset

Load

Save

Mémoire: 256 B

Dauphin Web (Aide)

Code

Assembler

Source

Code

```
;Hello_pixel
NBLETTERS = 11

TOP:

START:
    CLR    Y            ; en haut
    CLR    B            ; rang 1ère lettre
NEXT:
    CLR    X            ; à gauche
LETTER:
    PUSH   Y
    MOVE   B, Y
    MOVE   TABLE+{X}+{Y}, A    ; lettre à
afficher
    POP    Y

    CALL   _DrawChar    ; affiche une lettre

    INC    X            ; à droite
    COMP   #8, X        ; dernière position x
?
    JUMP,LO LETTER      ; non -> LETTER
                    ; oui ->
    MOVE   #20, A
    CALL   WAITSEC      ; attend 1
seconde

    INC    B            ; 1ère lettre suivante
    COMP   #7+NBLETTERS, B ; dernière lettre ?
    JUMP,LS NEXT        ; non -> NEXT
                    ; oui ->
    INC    Y            ; plus bas
    COMP   #5, Y        ; bas atteint ?
    JUMP,LO START       ; non -> START
    JUMP   TOP          ; oui -> TOP
```

CPU

A B X Y

Registres	A	B	X	Y	PC	SP
Valeur:	20	07	08	02	20	98

Fanions	N	Z	C	F
Valeur:	FALSE	FALSE	FALSE	FALSE

BUS

Read

Write

Adresse: 20000100100

Donnée: 25511111111

Mémoire (RAM)

hex

dec

35	33	34	11	71	84	48	39	15	255	00	40	42	114	08	20
00	03	80	20	255	00	00	41	113	18	22	00	02	43	115	05
20	00	01	16	00	00	26	32	32	32	32	32	32	68	97	
117	112	104	105	110	32	87	101	98	32	32	32	32	32	32	32
32	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	02	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
102	02	66	00	81	85	64	80	87	87	98	112	85	84	82	112
103	116	83	80	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Instructions: [Hide](#) Adresses mémoire: [Show](#)

Sorties

ASCII

Digital

Bitmap

0000

Dauphin

Interface



Horloge: 16 HZ ▶ Run ⏸ Step Turbo Reset Load Save Mémoire: 256 B ▼ Dauphin Web (Aide)

Code Assembler Source Code

```
;Hello_pixel
NBLETTERS = 11

TOP:

START:
  CLR    Y          ; en haut
  CLR    B          ; rang 1ère
NEXT:
  CLR    X          ; à gauche
LETTER:
  PUSH   Y
  MOVE   B, Y
  MOVE   TABLE+{X}+{Y}, A
afficher
  POP    Y

  CALL   _DrawChar  ; affiche u

  INC    X          ; à droite
  COMP   #8, X      ; dernière
?
  JUMP,LO LETTER    ; non -> LE
                  ; oui ->

  MOVE   #20, A
  CALL   WAITSEC    ; attend 1
seconde

  INC    B          ; 1ère lettre suivante
  COMP   #7+NBLETTERS, B ; dernière lettre ?
  JUMP,LS NEXT      ; non -> NEXT
                  ; oui ->
  INC    Y          ; plus bas
  COMP   #5, Y      ; bas atteint ?
  JUMP,LO START     ; non -> START
  JUMP   TOP        ; oui -> TOP
```

CPU

Registres	A	B	X	Y	PC	SP
Valeur:	20	07	08	02	20	98

Fanions	N	Z	C	F

Sorties

ASCII

Digital

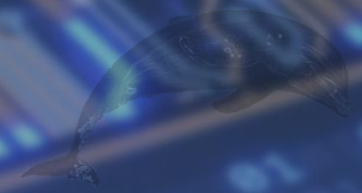
BUS Read Write

Adresse: 20 0 0 0 1 0 1 0 0

Donnée: 255 1 1 1 1 1 1 1 1

Instructions: [Hide](#) Adresses mémoire: [Show](#)

Interface



Horloge: 16 HZ

Run

Step

Turbo

Reset

Load

Save

Mémoire: 256 B

Dauphin Web (Aide)

Code

Assembler

Source

Code

```
;Hello_pixel
NBLETTERS = 11

TOP:

START:
  CLR    Y          ; en haut
  CLR    B          ; rang 1ère lettre
NEXT:
  CLR    X          ; à gauche
LETTER:
  PUSH   Y
  MOVE   B, Y
  MOVE   TABLE+{X}+{Y}, A      ; lettre à
afficher
  POP    Y

  CALL   _DrawChar      ; affiche une lettre

  INC    X              ; à droite
  COMP   #8, X          ; dernière position x
?
  JUMP,LO LETTER        ; non -> LETTER
                        ; oui ->
  MOVE   #20, A
  CALL   WAITSEC        ; attend 1
seconde

  INC    B              ; 1ère lettre suivante
  COMP   #7+NBLETTERS, B ; dernière lettre ?
  JUMP,LS NEXT          ; non -> NEXT
                        ; oui ->
  INC    Y              ; plus bas
  COMP   #5, Y          ; bas atteint ?
  JUMP,LO START         ; non -> START
  JUMP   TOP            ; oui -> TOP
```

CPU

A B X Y

Registres	A	B	X	Y	PC	SP
Valeur:	20	07	08	02	20	98

Fanions	N	Z	C	F
Valeur:	FALSE	FALSE	FALSE	FALSE

BUS

Read

Write

Adresse: 2000010100

Donnée: 2551111111

Mémoire (RAM)

hexdec

35	33	34	11	71	84	48	39	15	255	00	40	42	114	08	20
00	03	80	20	255	00	00	41	113	18	22	00	02	43	115	05
20	00	01	16	00	00	26	32	32	32	32	32	32	68	97	
117	112	104	105	110	32	87	101	98	32	32	32	32	32	32	32
32	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	02	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
102	02	66	00	81	85	64	80	87	87	98	112	85	84	82	112
103	116	83	80	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Instructions: [Hide](#) Adresses mémoire: [Show](#)

Sorties

ASCII

Digital

Bitmap

0000

Dauphin

Interface

Horloge: 16 HZ ▶ Run ▶▶ Step Turbo Reset Load Save Mémoire: 256 B Dauphin Web (Aide)

Code Assembler S

```
;Hello_pixel
NBLETTERS = 11

TOP:
    CLR    Y            ; en haut
START:
    CLR    B            ; rang 1ère
NEXT:
    CLR    X            ; à gauche
LETTER:
    PUSH   Y
    MOVE   B, Y
    MOVE   TABLE+{X}+{Y}, A    ;
afficher
    POP    Y
    CALL   _DrawChar    ; affiche u
    INC    X            ; à droite
    COMP   #8, X        ; dernière
    ?
    JUMP,LO LETTER      ; non -> LE
                        ; oui ->
    MOVE   #20, A
    CALL   WAITSEC      ; a
seconde

    INC    B            ; 1ère lett
    COMP   #7+NBLETTERS, B ; dernière
    JUMP,LS NEXT        ; non -> NE
                        ; oui ->
    INC    Y            ; plus bas
    COMP   #5, Y        ; bas atteint
    JUMP,LO START       ; non -> START
    JUMP   TOP          ; oui -> TOP
```

Mémoire (RAM) hex dec

35	33	34	11	71	84	48	39	15	255	00	40	42	114	08	20
00	03	80	20	255	00	00	41	113	18	22	00	02	43	115	05
20	00	01	16	00	00	26	32	32	32	32	32	32	32	68	97
117	112	104	105	110	32	87	101	98	32	32	32	32	32	32	32
32	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	02	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
102	02	66	00	81	85	64	80	87	87	98	112	85	84	82	112
103	116	83	80	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Instructions: [Hide](#) Adresses mémoire: [Show](#)

ASCII Digital Bitmap

0.

UPHIN

Instructions: [Hide](#) Adresses mémoire: [Show](#)

Interface

Horloge: 16 HZ Run Step Turbo Reset Load Save Mémoire: 256 B Dauphin Web (Aide)

Code Assembler Source Code

```
;Hello_pixel
NBLETTERS = 11

TOP:
    CLR    Y        ; en haut
START:
    CLR    B        ; rang ière lettre
```

Mémoire (RAM) hex dec

35	33	34	11	71	84	48	39	15	255	00	40	42	114	08	20
00	03	80	20	255	00	00	41	113	18	22	00	02	43	115	05
20	00	01	16	00	00	26	32	32	32	32	32	32	32	68	97
117	112	104	105	110	32	87	101	98	32	32	32	32	32	32	32
32	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	02	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
102	02	66	00	81	85	64	80	87	87	98	112	85	84	82	112
103	116	83	80	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Instructions: [Hide](#) Adresses mémoire: [Show](#)

```
JUMP,LS NEXT    ; non -> NEXT
                ; oui ->
INC    Y        ; plus bas
COMP   #5, Y     ; bas atteint ?
JUMP,LO START    ; non -> START
JUMP   TOP       ; oui -> TOP
```

CPU A B X Y

Sorties ASCII Digital Bitmap

Mémoire (RAM) hex dec

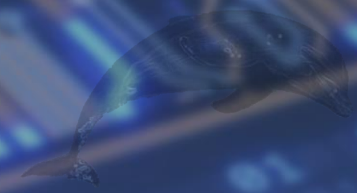
00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
35	33	34	11	71	84	48	39	15	255	00	40	42	114	08	20
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
00	03	80	20	255	00	00	41	113	18	22	00	02	43	115	05
32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
20	00	01	16	00	00	26	32	32	32	32	32	32	32	68	97
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
117	112	104	105	110	32	87	101	98	32	32	32	32	32	32	32
64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
32	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
00	00	02	00	00	00	00	00	00	00	00	00	00	00	00	00
112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
102	02	66	00	81	85	64	80	87	87	98	112	85	84	82	112
192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
103	116	83	80	00	00	00	00	00	00	00	00	00	00	00	00
208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Instructions: [Hide](#) Adresses mémoire: [Show](#)

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Instructions: [Hide](#) Adresses mémoire: [Show](#)

Interface



Horloge: 16 HZ

Run

Step

Turbo

Reset

Load

Save

Mémoire: 256 B

Dauphin Web (Aide)

Code

Assembler

Source

Code

```
;Hello_pixel
NBLETTERS = 11

TOP:

    CLR    Y            ; en haut
START:
    CLR    B            ; rang 1ère lettre
NEXT:
    CLR    X            ; à gauche
LETTER:
    PUSH   Y
    MOVE   B, Y
    MOVE   TABLE+(X)+{Y}, A    ; lettre à
afficher
    POP    Y

    CALL   _DrawChar    ; affiche une lettre

    INC    X            ; à droite
    COMP   #8, X        ; dernière position x
?
    JUMP,LO LETTER      ; non -> LETTER
                        ; oui ->
    MOVE   #20, A
    CALL   WAITSEC      ; attend 1
seconde

    INC    B            ; 1ère lettre suivante
    COMP   #7+NBLETTERS, B ; dernière lettre ?
    JUMP,LS NEXT        ; non -> NEXT
                        ; oui ->
    INC    Y            ; plus bas
    COMP   #5, Y        ; bas atteint ?
    JUMP,LO START       ; non -> START
    JUMP   TOP          ; oui -> TOP
```

CPU

A B X Y

Registres	A	B	X	Y	PC	SP
Valeur:	20	07	08	02	20	98

Fanions	N	Z	C	F
Valeur:	FALSE	FALSE	FALSE	FALSE

BUS

Read

Write

Adresse: 20 0 0 0 1 0 1 0 0

Donnée: 255 1 1 1 1 1 1 1 1

Mémoire (RAM)

hex dec

35	33	34	11	71	84	48	39	15	255	00	40	42	114	08	20
00	03	80	20	255	00	00	41	113	18	22	00	02	43	115	05
20	00	01	16	00	00	26	32	32	32	32	32	32	68	97	
117	112	104	105	110	32	87	101	98	32	32	32	32	32	32	32
32	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	02	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
102	02	66	00	81	85	64	80	87	87	98	112	85	84	82	112
103	116	83	80	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Instructions: [Hide](#) Adresses mémoire: [Show](#)

Sorties

ASCII

Digital

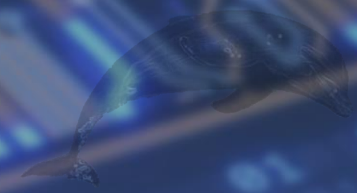
Bitmap

0.0.0.0

Dauphin

Instructions: [Hide](#) Adresses mémoire: [Show](#)

Interface



Horloge: 16 HZ

Run

Step

Turbo

Reset

Load

Save

Mémoire: 256 B

Dauphin Web (Aide)

Code

Assembler

Source

Code

```
;Hello_pixel
NBLETTERS = 11

TOP:

    CLR    Y            ; en haut
START:
    CLR    B            ; rang 1ère lettre
NEXT:
    CLR    X            ; à gauche
LETTER:
    PUSH   Y
    MOVE   B, Y
    MOVE   TABLE+(X)+(Y), A    ; lettre à
afficher
    POP    Y

    CALL   _DrawChar    ; affiche une lettre

    INC    X            ; à droite
    COMP   #8, X        ; dernière position x
?
    JUMP,LO LETTER      ; non -> LETTER
                        ; oui ->
    MOVE   #20, A
    CALL   WAITSEC      ; attend 1
seconde

    INC    B            ; 1ère lettre suivante
    COMP   #7+NBLETTERS, B ; dernière lettre ?
    JUMP,LS NEXT        ; non -> NEXT
                        ; oui ->
    INC    Y            ; plus bas
    COMP   #5, Y        ; bas atteint ?
    JUMP,LO START       ; non -> START
    JUMP   TOP          ; oui -> TOP
```

CPU

A

B

X

Y

Registres	A	B	X	Y	PC	SP
Valeur:	20	07	08	02	20	98

Fanions	N	Z	C	F
Valeur:	FALSE	FALSE	FALSE	FALSE

BUS

Read

Write

Adresse: 20

0 0 0 1 0 1 0 0

Donnée: 255

1 1 1 1 1 1 1 1

Mémoire (RAM)

hex

dec

35	33	34	11	71	84	48	39	15	255	00	40	42	114	08	20
00	03	80	20	255	00	00	41	113	18	22	00	02	43	115	05
20	00	01	16	00	00	26	32	32	32	32	32	32	68	97	
117	112	104	105	110	32	87	101	98	32	32	32	32	32	32	32
32	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	02	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
102	02	66	00	81	85	64	80	87	87	98	112	85	84	82	112
103	116	83	80	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Instructions: [Hide](#) Adresses mémoire: [Show](#)

Sorties

ASCII

Digital

Bitmap

0000

Dauphin

Interface

Horloge: 16 HZ ▶ Run ▶▶ Step Turbo Reset Load Save Mémoire: 256 B Dauphin Web (Aide)

Code Assembler Source Code

```
;Hello_pixel
NBLETTERS = 11

TOP:
    CLR    Y        ; en haut
START:
    CLR    B        ; rang 1ère
NEXT:
    CLR    X        ; à gauche
LETTER:
    PUSH   Y
    MOVE   B, Y
    MOVE   TABLE+(X)+(Y), A    ;
afficher
    POP    Y

    CALL   _DrawChar    ; affiche u

    INC    X        ; à droite
    COMP   #8, X      ; dernière
?
    JUMP,LO LETTER    ; non -> LE
                    ; oui ->
    MOVE   #20, A
    CALL   WAITSEC    ; attend 1
seconde

    INC    B        ; 1ère lettre suivante
    COMP   #7+NBLETTERS, B    ; dernière lettre ?
    JUMP,LS NEXT      ; non -> NEXT
                    ; oui ->
    INC    Y        ; plus bas
    COMP   #5, Y      ; bas atteint ?
    JUMP,LO START     ; non -> START
    JUMP   TOP        ; oui -> TOP
```

CPU

A B X Y

Registres	A	B	X	Y	PC	SP
Valeur:	20	07	08	02	20	98

Fanions	N	Z	C	F
Valeur:	FALSE	FALSE	FALSE	FALSE

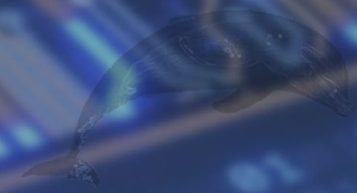
Sorties ASCII Digital Bitmap

ASCII



Instructions: [Hide](#) Adresses mémoire: [Show](#)

Interface



Horloge: 16 HZ

Run

Step

Turbo

Reset

Load

Save

Mémoire: 256 B

Dauphin Web (Aide)

Code

Assembler

Source

Code

```
;Hello_pixel
NBLETTERS = 11

TOP:

START:
    CLR    Y            ; en haut
    CLR    B            ; rang 1ère lettre
NEXT:
    CLR    X            ; à gauche
LETTER:
    PUSH   Y
    MOVE   B, Y
    MOVE   TABLE+{X}+{Y}, A    ; lettre à
afficher
    POP    Y

    CALL   _DrawChar    ; affiche une lettre

    INC    X            ; à droite
    COMP   #8, X        ; dernière position x
?
    JUMP,LO LETTER      ; non -> LETTER
                    ; oui ->
    MOVE   #20, A
    CALL   WAITSEC      ; attend 1
seconde

    INC    B            ; 1ère lettre suivante
    COMP   #7+NBLETTERS, B ; dernière lettre ?
    JUMP,LS NEXT        ; non -> NEXT
                    ; oui ->
    INC    Y            ; plus bas
    COMP   #5, Y        ; bas atteint ?
    JUMP,LO START       ; non -> START
    JUMP   TOP          ; oui -> TOP
```

CPU

A B X Y

Registres	A	B	X	Y	PC	SP
Valeur:	20	07	08	02	20	98

Fanions	N	Z	C	F
Valeur:	FALSE	FALSE	FALSE	FALSE

BUS

Read

Write

Adresse: 20

0 0 0 1 0 1 0 0

Donnée: 255

1 1 1 1 1 1 1 1

Mémoire (RAM)

hex

dec

35	33	34	11	71	84	48	39	15	255	00	40	42	114	08	20
00	03	80	20	255	00	00	41	113	18	22	00	02	43	115	05
20	00	01	16	00	00	26	32	32	32	32	32	32	68	97	
117	112	104	105	110	32	87	101	98	32	32	32	32	32	32	32
32	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	02	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
102	02	66	00	81	85	64	80	87	87	98	112	85	84	82	112
103	116	83	80	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Instructions: [Hide](#) Adresses mémoire: [Show](#)

Sorties

ASCII

Digital

Bitmap

0.0.0.0

Interface

Horloge: 16 HZ ▶ Run ▶▶ Step Turbo

Code Assembler Source

```
;Hello_pixel
NBLETTERS = 11

TOP:
    CLR    Y            ; en haut
START:
    CLR    B            ; rang 1ère lettre
NEXT:
    CLR    X            ; à gauche
LETTER:
    PUSH   Y
    MOVE   B, Y
    MOVE   TABLE+(X)+(Y), A    ; lettre à
afficher
    POP    Y

    CALL   _DrawChar    ; affiche une lettre

    INC    X            ; à droite
    COMP   #8, X        ; dernière position
?
    JUMP,LO LETTER      ; non -> LETTER
                    ; oui ->
    MOVE   #20, A
    CALL   WAITSEC      ; attend 1
seconde

    INC    B            ; 1ère lettre suiv
    COMP   #7+NBLETTERS, B ; dernière lettre ?
    JUMP,LS NEXT        ; non -> NEXT
                    ; oui ->
    INC    Y            ; plus bas
    COMP   #5, Y        ; bas atteint ?
    JUMP,LO START       ; non -> START
    JUMP   TOP          ; oui -> TOP
```

Sorties

ASCII Digital Bitmap

ASCII

Digital

Bitmap

Dauphin Web (Aide)

ASCII Digital Bitmap

Digital

Bitmap

Instructions: Hide Adresses mémoire: Show

Interface

Horloge: 16 HZ ▶ Run ▶▶ Step Turbo Reset Load Save Mémoire: 256 B Dauphin Web (Aide)

Code Ass CPU A B X Y

Sorties ASCII Digital Bitmap

ASCII

Digital

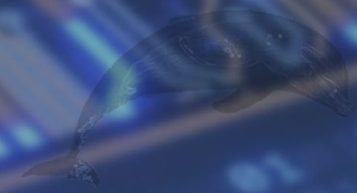
Bitmap

Sorties ASCII Digital Bitmap

Bitmap

Instructions: [Hide](#) Adresses mémoire: [Show](#)

Interface



Horloge: 16 HZ

Run

Step

Turbo

Reset

Load

Save

Mémoire: 256 B

Dauphin Web (Aide)

Code

Assembler

Source

Code

```
;Hello_pixel
NBLETTERS = 11

TOP:

CLR      Y            ; en haut
START:
CLR      B            ; rang 1ère lettre
NEXT:
CLR      X            ; à gauche
LETTER:
PUSH     Y
MOVE     B, Y
MOVE     TABLE+{X}+{Y}, A      ; lettre à
afficher
POP      Y

CALL     DrawChar      ; affiche une lettre

INC      X            ; à droite
COMP     #8, X         ; dernière position x
?
JUMP,LO  LETTER       ; non -> LETTER
; oui ->

MOVE     #20, A
CALL     WAITSEC      ; attend 1
seconde

INC      B            ; 1ère lettre suivante
COMP     #7+NBLETTERS, B ; dernière lettre ?
JUMP,LS  NEXT         ; non -> NEXT
; oui ->

INC      Y            ; plus bas
COMP     #5, Y         ; bas atteint ?
JUMP,LO  START        ; non -> START
JUMP     TOP          ; oui -> TOP
```

CPU

A B X Y

Registres	A	B	X	Y	PC	SP
Valeur:	20	07	08	02	20	98

Fanions	N	Z	C	F
Valeur:	FALSE	FALSE	FALSE	FALSE

BUS

Read

Write

Adresse: 2000010100

Donnée: 2551111111

Mémoire (RAM)

hex

dec

35	33	34	11	71	84	48	39	15	255	00	40	42	114	08	20
00	03	80	20	255	00	00	41	113	18	22	00	02	43	115	05
20	00	01	16	00	00	26	32	32	32	32	32	32	68	97	
117	112	104	105	110	32	87	101	98	32	32	32	32	32	32	
32	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00	00	02	00	00	00	00	00	00	00	00	00	00	00	00	
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
102	02	66	00	81	85	64	80	87	87	98	112	85	84	82	112
103	116	83	80	00	00	00	00	00	00	00	00	00	00	00	
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	

Instructions: [Hide](#) Adresses mémoire: [Show](#)

Sorties

ASCII

Digital

Bitmap

0.0.0.0

Interface

Horloge: 16 HZ ▶ Run ▶▶ Step Turbo Minimise Close Dauphin Web (Aide)

Code Assembler Source Code

```
;Hello_pixel
NBLETTERS = 11

TOP:
    CLR    Y            ; en haut
START:
    CLR    B            ; rang 1ère lettre
NEXT:
    CLR    X            ; à gauche
LETTER:
    PUSH   Y
    MOVE   B, Y
    MOVE   TABLE+{X}+{Y}, A    ; lettre à
afficher
    POP    Y

    CALL   _DrawChar    ; affiche une lettre

    INC    X            ; à droite
    COMP   #8, X        ; dernière position x
?
    JUMP,LO LETTER      ; non -> LETTER
                    ; oui ->
    MOVE   #20, A
    CALL   WAITSEC      ; attend 1
seconde

    INC    B            ; 1ère lettre suivante
    COMP   #7+NBLETTERS, B ; dernière lettre ?
    JUMP,LS NEXT        ; non -> NEXT
                    ; oui ->
    INC    Y            ; plus bas
    COMP   #5, Y        ; bas atteint ?
    JUMP,LO START       ; non -> START
    JUMP   TOP          ; oui -> TOP
```

Code Assembler Source Code

```
;Hello_pixel
NBLETTERS = 11

TOP:
    CLR    Y            ; en haut
START:
    CLR    B            ; rang 1ère lettre
NEXT:
    CLR    X            ; à gauche
LETTER:
    PUSH   Y
    MOVE   B, Y
    MOVE   TABLE+{X}+{Y}, A    ; lettre à
afficher
    POP    Y

    CALL   _DrawChar    ; affiche une lettre

    INC    X            ; à droite
    COMP   #8, X        ; dernière position x
?
    JUMP,LO LETTER      ; non -> LETTER
                    ; oui ->
    MOVE   #20, A
    CALL   WAITSEC      ; attend 1
seconde

    INC    B            ; 1ère lettre suivante
    COMP   #7+NBLETTERS, B ; dernière lettre ?
    JUMP,LS NEXT        ; non -> NEXT
                    ; oui ->
    INC    Y            ; plus bas
    COMP   #5, Y        ; bas atteint ?
    JUMP,LO START       ; non -> START
    JUMP   TOP          ; oui -> TOP
```

ASCII Digital Bitmap

8.8.8.

dauphin

Instructions: Hide Adresses mémoire: Show

Interface

CodeAssembleurSourceCode

```

;Hello_pixel
NBLETTERS = 11

TOP:
START: CLR Y ; en haut
NEXT: CLR B ; rang 1ère lettre
LETTER: CLR X ; à gauche
        PUSH Y
        MOVE B, Y
        MOVE TABLE+{X}+{Y}, A ; lettre à
afficher POP Y
        CALL _DrawChar ; affiche une lettre
        INC X ; à droite
        COMP #8, X ; dernière position x
?
        JUMP,LO LETTER ; non -> LETTER
        ; oui ->
        MOVE #20, A
        CALL _WAITSEC ; attend 1
seconde
        INC B ; 1ère lettre suivante
        COMP #7+NBLETTERS, B ; dernière lettre ?
        JUMP,LS NEXT ; non -> NEXT
        ; oui ->
        INC Y ; plus bas
        COMP #5, Y ; bas atteint ?
        JUMP,LO START ; non -> START
        JUMP TOP ; oui -> TOP

```

CodeAssembleurSourceCode

Adr.	Valeur	Instruction
00	35	CLR Y ; en haut
01	33	CLR B ; rang 1ère lettre
02	34	CLR X ; à gauche
03	11	PUSH Y
04	71	MOVE B, Y
05	84 48 39	MOVE TABLE+{X}+{Y}, A ; lettre à afficher
08	15	POP Y
09	255 00 40	CALL _DrawChar ; affiche une lettre
12	42	INC X ; à droite
13	114 08	COMP #8, X ; dernière position x ?
15	20 00 03	JUMP,LO LETTER ; non -> LETTER
18	80 20	MOVE #20, A
20	255 00 00	CALL _WAITSEC ; attend 1 seconde
23	41	INC B ; 1ère lettre suivante
24	113 18	COMP #7+NBLETTERS, B ; dernière lettre ?



Discussion

Dauphin Web vs. Simulateur de dauphin

Validation

Améliorations

Dauphin Web vs. Simulateur de dauphin

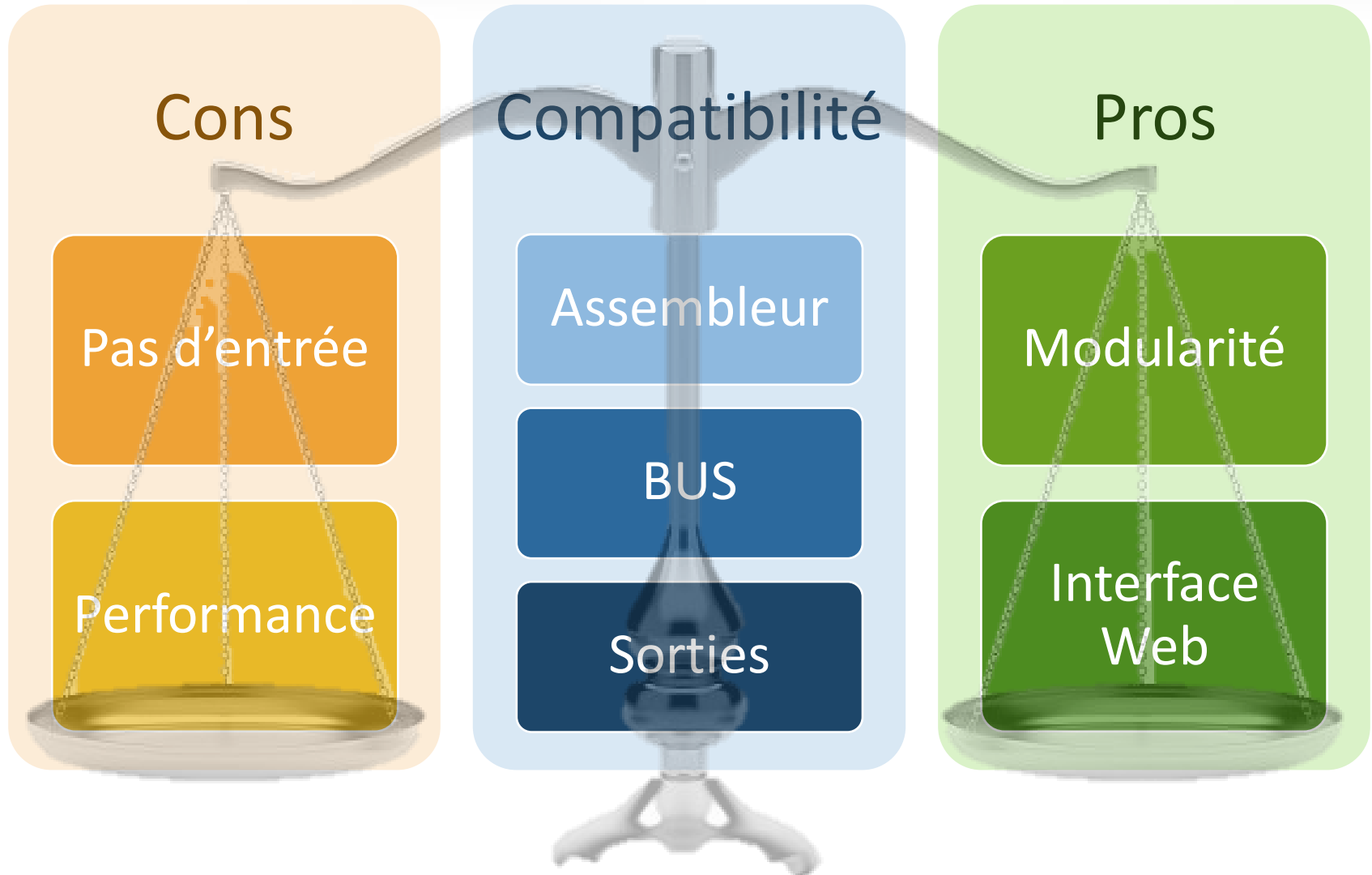
Compatibilité

Assembleur

BUS

Sorties

Dauphin Web vs. Simulateur de dauphin



Validation



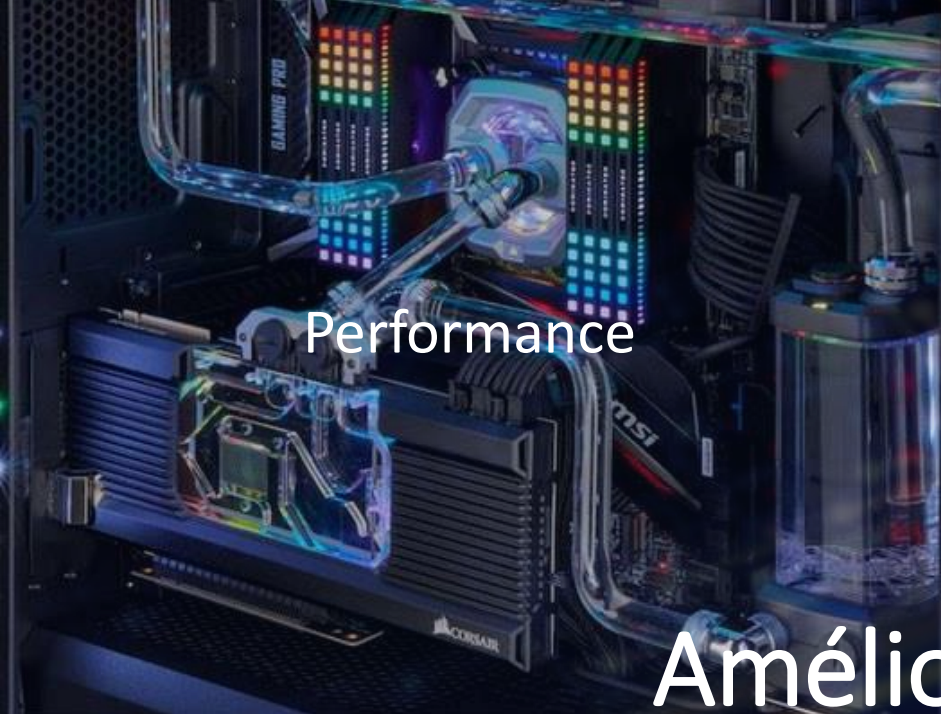
Fonctionnement

Tests extensifs de toutes les commandes



Outil pédagogique

Hérité de son aîné



Activités

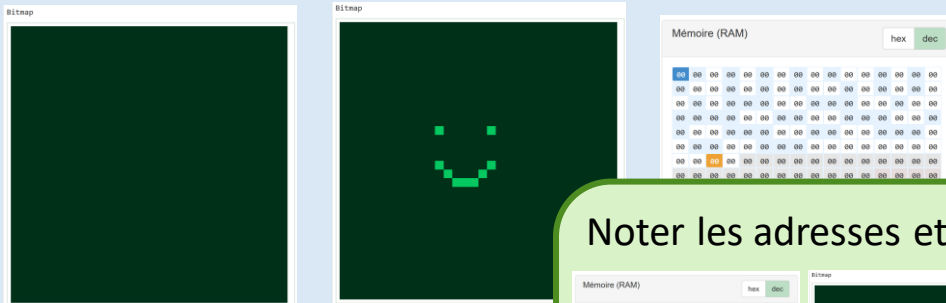


https://kunznicolas.github.io/Web_Dauphin/

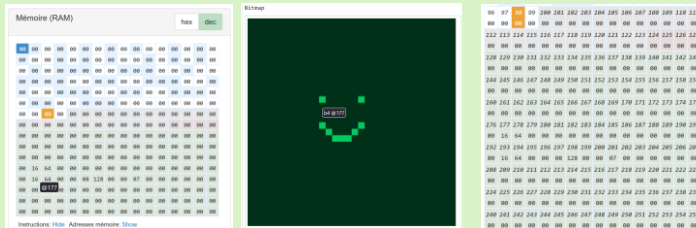
Pour une expérience optimale: min 1200px résolution horizontale

Activité 1 – affichage / mémoire / bus

Dessiner sur l'écran Bitmap
Observer les changements dans la mémoire



Noter les adresses et valeurs mémoires



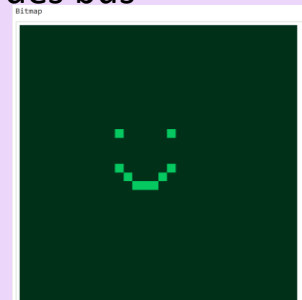
Echanger vos notes
Inscrire en mémoire les valeurs à l'aides des bus



BUS

Adresse: 177

Donnée: 16



Conclusion



Dauphin Web:
Simulateur d'ordinateur 8bits

Application web
Open-source ([GitHub](#))
Compatible avec le simulateur de l'EPSITEC



Outils déjà établis comme atout pédagogique

Test en classe



Problème de performance

Remerciements

- Biljana Petreska von Ritter-Zahony (HEP Vaud)
- Jean-Philippe Pellet (HEP Vaud)
- Michel Favre (Gymnase d'Yverdon)
- Laurent Michel (Gymnase de Beaulieu)
- Pierre Arnaud (et l'EPSITEC)
- Andreas Humm (UNIFR, GymInf)
- Mélissa Nicolier et Arthur Dromard (Gymnase d'Yverdon, GymInf)
- Yohan Sanchez (Gymnase d'Yverdon, correcteur)

The background of the slide is a close-up, angled view of a circuit board. It features intricate patterns of glowing blue and orange lines, representing data paths or circuit traces. In the upper right corner, there is a rectangular area displaying binary code (0s and 1s) in a glowing blue font. The overall aesthetic is high-tech and digital.

Retour sur l'activité

