

Électronique

Arduino

Séance 2



Plan

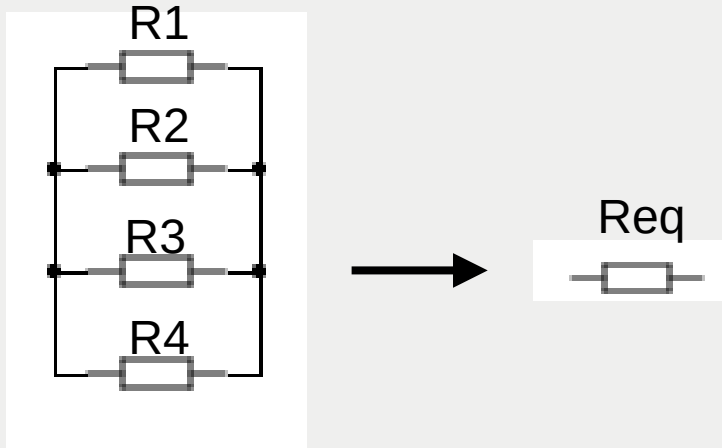
- Électronique et composants
- Arduino et programmation
- Application pratique



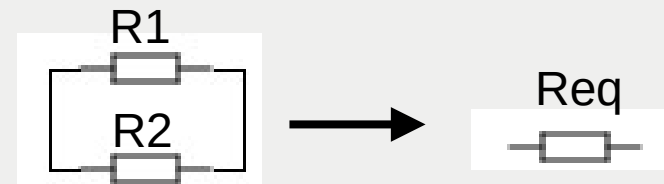
Électronique et composants



Association de résistances



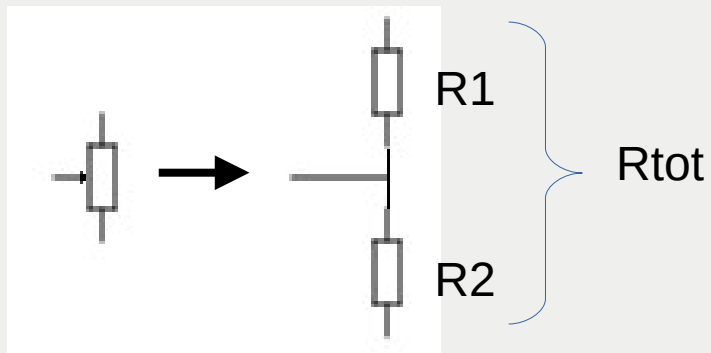
$$\frac{1}{Req} = \frac{1}{R1} + \frac{1}{R2} + \frac{1}{R3} + \frac{1}{R4}$$



$$Req = \frac{R1 \times R2}{R1 + R2}$$



Potentiomètre



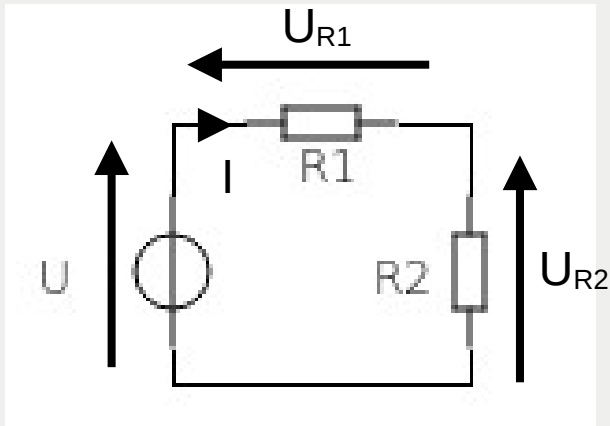
$$R_{\text{tot}} = R1 + R2$$

$$R1 = R_{\text{tot}} - R2$$

$$R2 = R_{\text{tot}} - R1$$



Pont diviseur de tension



Loi d'Ohm :

$$U = R_{eq} \times I = (R_1 + R_2) \times I$$

$$\Rightarrow I = U \times 1/(R_1 + R_2)$$

$$U_{R2} = R_2 \times I$$

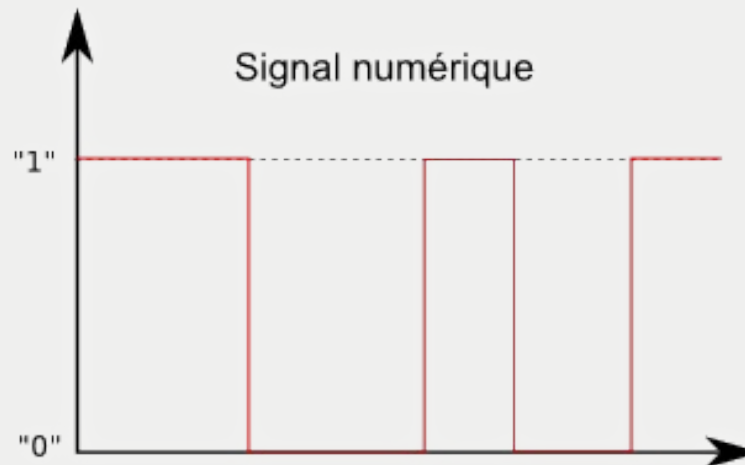
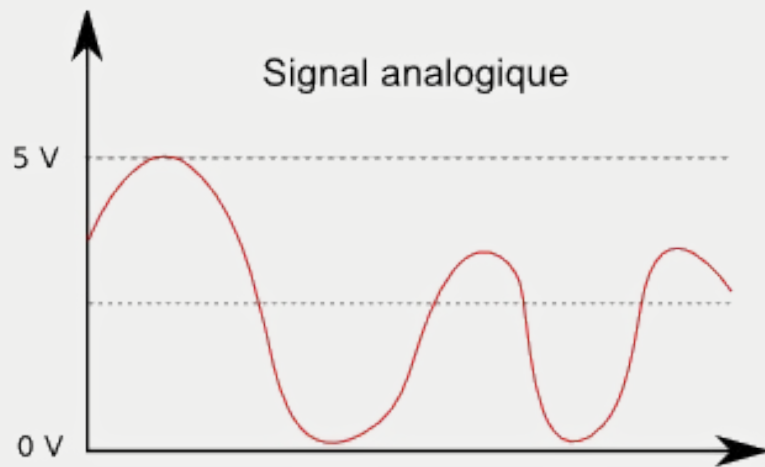
$$U_{R2} = U \times R_2 / (R_1 + R_2)$$



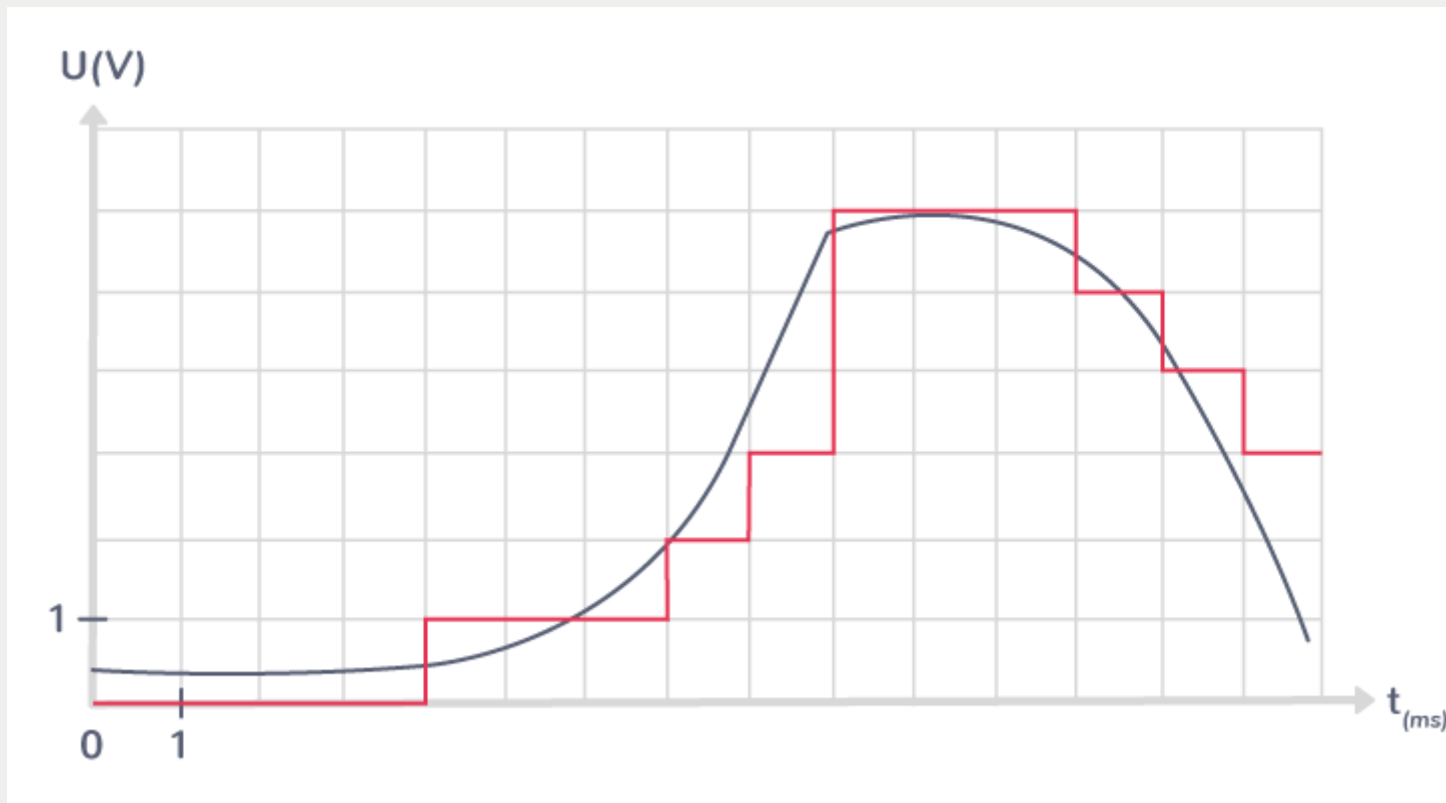
Des questions ?



Analogique vs numérique



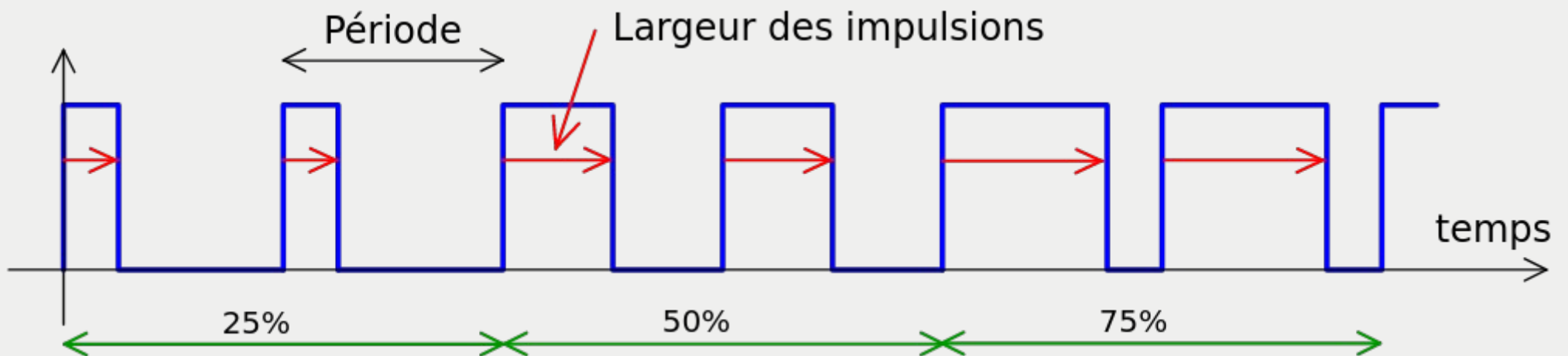
Conversion $A \leftrightarrow N$



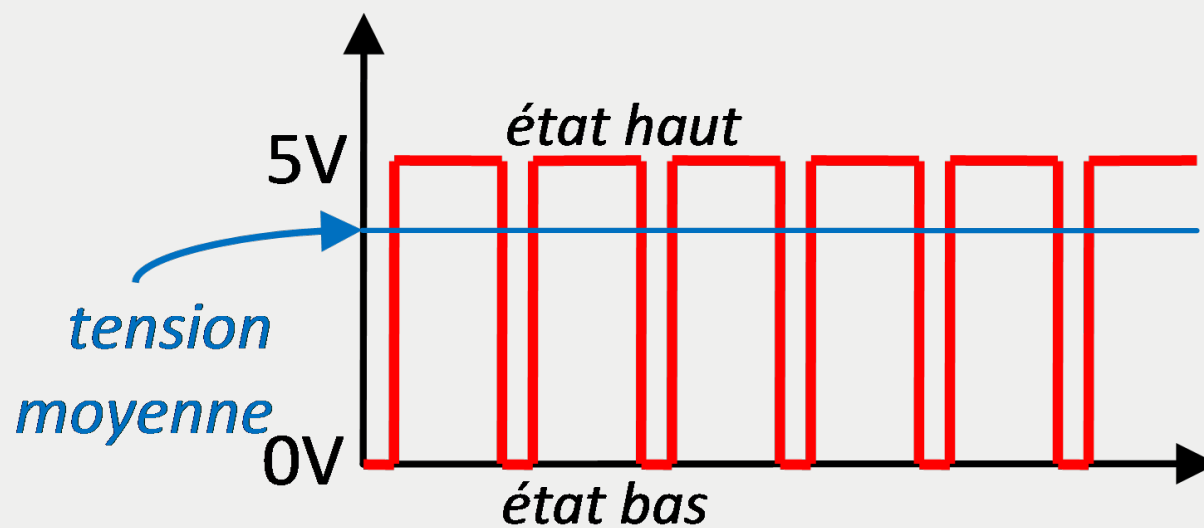
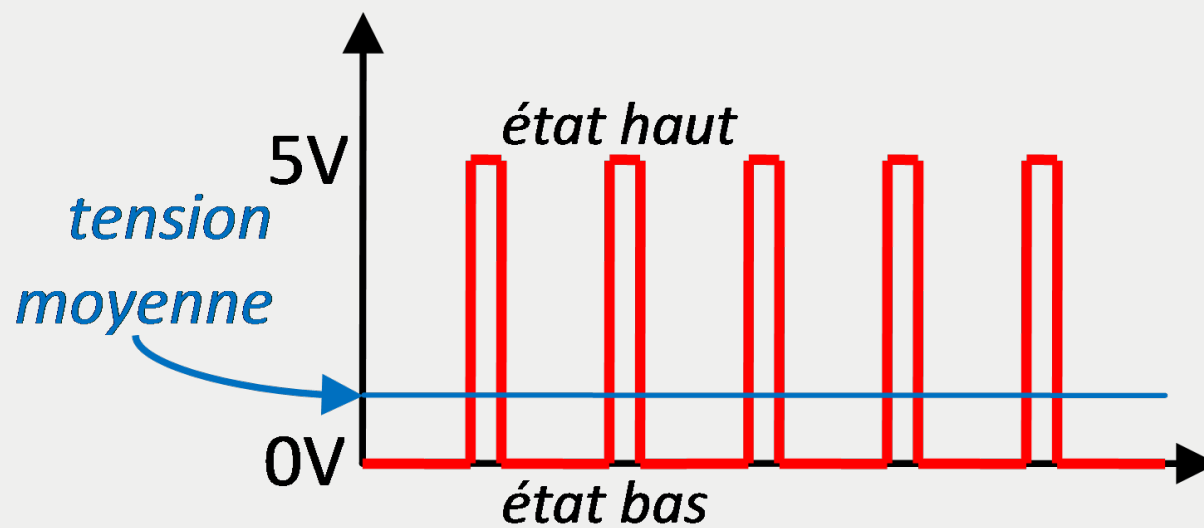
PWM ou MLI

Pulse Width Modulation

Modulation à Largeur d'Impulsions



PWM



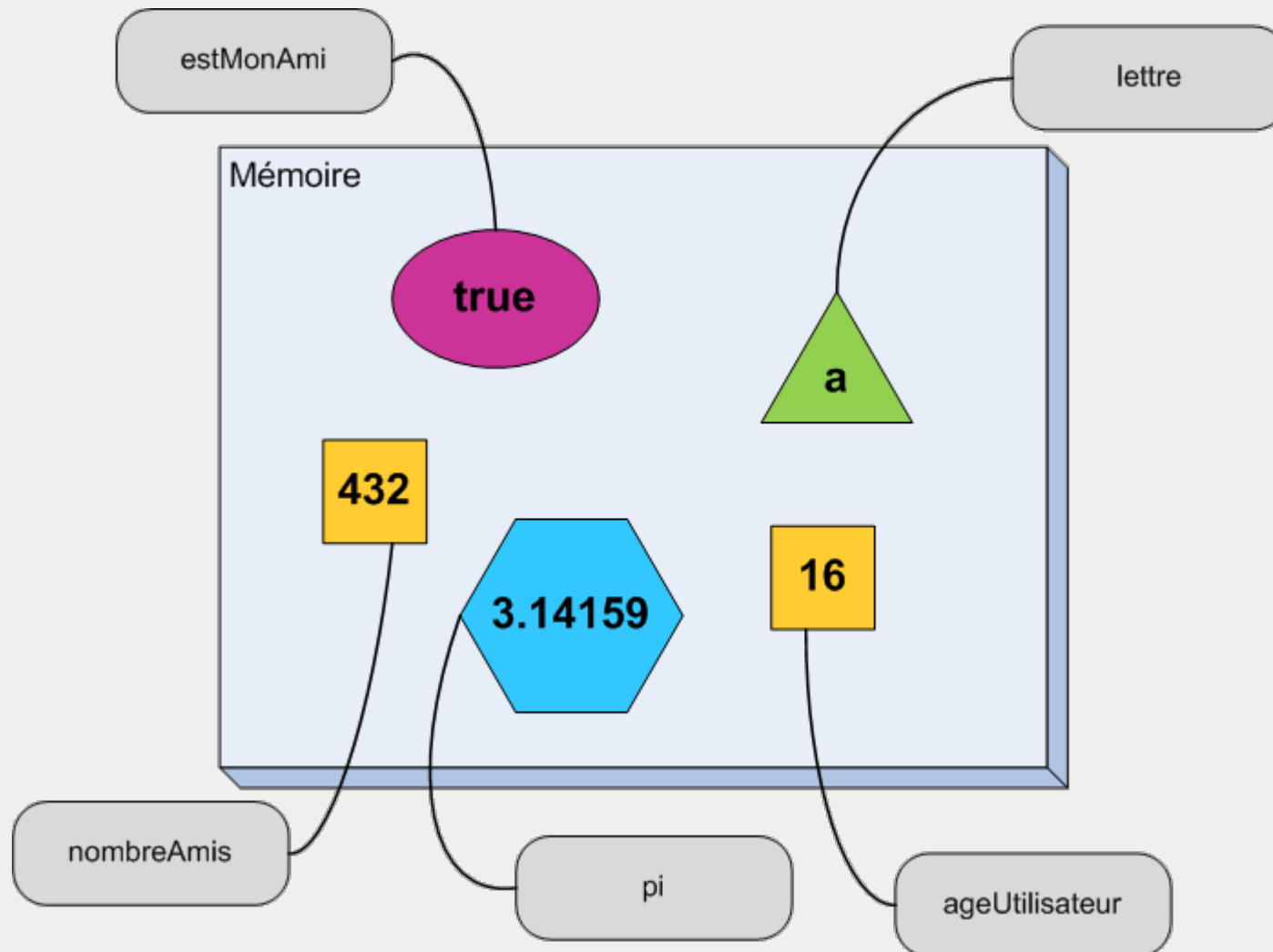
Des questions ?



Arduino et programmation



Mémoire et variables



Types de variables

Type	Définition	Valeur min	Valeur max
int	Nombre entier	-32 768	32 767
unsigned int	Nombre entier positif	0	65 535
long	Nombre entier	-2 147 483 648	2 147 483 647
unsigned long	Nombre entier positif	0	4 294 967 295
float	Nombre à virgule	$-3.4028235 \times 10^{38}$	3.4028235×10^{38}
bool	Faux / Vrai	false	true
char	Caractère	-128	127
unsigned char	Caractère	0	255
const type	Constante	Dépend du type	Dépend du type



Utilisation des variables

```
1 int x = 24;
2 int y; // y = ?????
3
4 void setup() {
5     y = 32;
6
7     // a et b n'existent pas
8     int a = 3;
9     // a existe mais pas b
10
11     int b = a;
12     // a et b existent et b = 3
13 }
14
15 void loop() {
16     // z n'existe pas
17     // a et b n'existent pas
18
19     int z;
20     // z existe mais z = ??????
21
22     z = y;
23     // z = 32
24 }
```



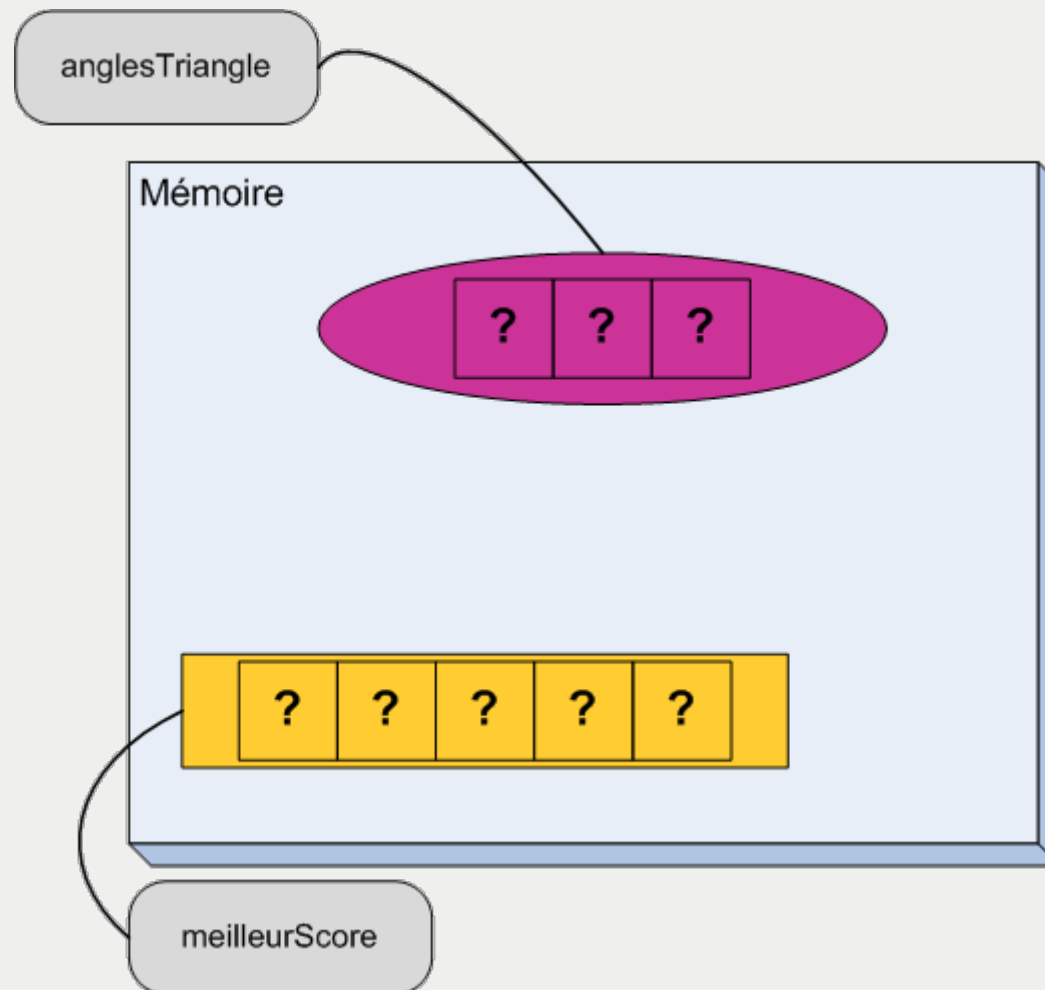
Utilisation des variables

```
1 int x = 24;
2 int y; // y = ?????
3
4 void setup() {
5     y = 32;
6
7     // a et b n'existent pas
8     int a = 3;
9     // a existe mais pas b
10
11     int b = a;
12     // a et b existent et b = 3
13 }
14
15 void loop() {
16     // z n'existe pas
17     // a et b n'existent pas
18
19     int z;
20     // z existe mais z = ??????
21
22     z = y;
23     // z = 32
24 }
```

```
1 const int led = 2;
2 const int led2 = 3;
3 const int temps = 1000;
4 // .....
5
6 void setup() {
7     pinMode(led, OUTPUT);
8     pinMode(led2, OUTPUT);
9 }
10
11 void loop() {
12     digitalWrite(led, LOW);
13     digitalWrite(led2, HIGH);
14     delay(temps);
15     digitalWrite(led, HIGH);
16     digitalWrite(led2, LOW);
17     delay(temps);
18 }
```



Tableaux



Utilisation des tableaux

```
1 int x[] = {24, 12, 13};
2 int y; // y = ?????
3
4 void setup() {
5     y = x[1];
6     // y = 12
7
8     // a et b n'existent pas
9     int a = x[0];
10    // a existe mais pas b et a = 24
11
12    int b = a;
13    // a et b existent et b = 24
14 }
15
16 void loop() {
17    // z n'existe pas
18    // a et b n'existent pas
19
20    int z[3];
21    // z existe mais z = {??????, ?????, ?????}
22
23    z[0] = x[2];
24    z[1] = x[1];
25    z[2] = x[0];
26    // z = {13, 12, 24}
27 }
28
29
```

```
1 const int leds[] = {2, 3};
2 const int temps = 1000;
3 // .....
4
5 void setup() {
6     pinMode(leds[0], OUTPUT);
7     pinMode(leds[1], OUTPUT);
8 }
9
10 void loop() {
11     digitalWrite(leds[0], LOW);
12     digitalWrite(leds[1], HIGH);
13     delay(temps);
14     digitalWrite(leds[0], HIGH);
15     digitalWrite(leds[1], LOW);
16     delay(temps);
17 }
18
```



Opérations sur les variables

```
1 int x = 31;  
2 int y = 5;  
3  
4 void setup() {  
5     int a;  
6  
7     a = x + y; // a = 36  
8  
9     a = x - y; // a = 26  
10  
11    a = x * y; // a = 155  
12  
13    a = x / y; // a = 6  
14    a = x % y; // a = 1  
15 }  
16
```

```
17 void loop() {  
18     int a;  
19     a = 1;  
20  
21     a += 2; // a = 3  
22  
23     a++; // a = 4  
24     a--; // a = 3  
25  
26     a *= 3; // a = 9  
27  
28     a -= 1; // a = 8  
29  
30     a /= 2; // a = 4  
31  
32     a %= 3; // a = 1  
33 }
```



Structures itératives

Tant que **condition**

action 1
action 2
action n

Fin Tant que

Algorithmique

while (**condition**)

```
{  
    action 1;  
    action 2;  
    action n;  
}
```

C/C++

Faire

action 1
action 2
action n

Tant que **condition**

Algorithmique

do

```
{  
    action 1;  
    action 2;  
    action n;  
}
```

while(**condition**);

C/C++

Pour (**cpt=0; cpt<10; cpt=cpt+1**)

action 1
action 2
action n

Fin Pour

Algorithmique

for (**int cpt=0; cpt<10; cpt++**)

{

Initialisation de
la variable de
comptage

Condition de
comptage

Pas de
comptage

action 1;
action 2;
action n;

}

C/C++



Structures itératives

```
1 const int leds[] = {2, 3, 4};
2 const int temps = 500;
3
4 void setup() {
5     while (i < 3) {
6         pinMode(leds[i], OUTPUT);
7
8         i++; // i = i + 1 ou i += 1
9     }
10 }
11
12 void loop() {
13     int i = 0;
14
15     do {
16         digitalWrite(leds[i], LOW);
17
18         i++;
19     } while (i < 3);
20
21     delay(500);
22
23     for(int i = 0; i < 3; i++) {
24         digitalWrite(leds[i], HIGH);
25     }
26
27     delay(500);
28 }
29
30
```



Des questions ?



Entrées analogiques

Lecture des entrées analogiques :

```
int x = analogRead(pin);
```

pin : numéro de la pin utilisée (De A0 à A5)

Valeur obtenue dans **x** : De 0 à 1023



Sorties PWM

Utilisation des sorties PWM (~) :
*analogWrite(**pin**, **valeur**);*

pin : numéro de la pin utilisée (3, 5, 6, 9, 10, 11)

valeur : valeur en sortie de la pin (De 0 à 255)



Des questions ?



Application pratique



Application pratique

Pour chaque niveau : faire le schéma, le montage et la programmation

- **Niveau 1**

- Contrôler une LED avec un potentiomètre

- **Niveau 2**

- Contrôler plusieurs LED avec un potentiomètre
- Stocker les pins dans un tableau et utiliser une boucle

- **Niveau 3**

- Faire un dégradé de couleurs sur plusieurs LED
- Contrôler l'intensité globale avec un potentiomètre

