

## EduC – Fonctions de lib EduC.h

En rouge, les compléments sont sous [www.didel.com/educ/EduC-NotesFonctions.pdf](http://www.didel.com/educ/EduC-NotesFonctions.pdf)

**EduC.h** regroupe les macros et fonctions liées à la carte. Le setup est **SetupEduC()**;

**Del60us()**; **Del200us()**; Deux délais de 60 et ~200 microsecondes, utilisés pour la mise au point seulement. – Enlevé des dernières versions

### Note 01

**DelMs (v16)**; Délai bloquant de 1 à 65000 millisecondes.

Synonymes: **delai (v16)**; (nom Arduino qui ne respecte pas la convention qui veut que les noms de fonctions commencent par une majuscule)

Exemples: **DelMs(100)**; **DelMs(att)**; avec **att** déclaré comme **byte att**; ou **byte att=100**; (max 255) ou comme **int att**: (max 65535)

## LEDs

**Leds with On/Off control**\_ - ce sont des macros, pas de () devant le ;

Note 02

**LedGOn**; **LedGOff**; **LedGToggle**; **LedDOn**; **LedDOff**; **LedDToggle**;

**RougeOn**; **RougeOff**; **VertOn**; **VertOff**; **BleuOn**; **BleuOff**; **BlancOn**; **BlancOff**;

**Leds with log control** - valeur 0 à 255 converties en PWM selon loi ~log (type byte)

Note 03

**LedG (v8)**; **LedR (v8)**; **Rouge (v8)**; **Vert(v8)**; **Bleu(v8)**;

**Leds with linear control** - Valeurs 0 à 255 converties en PWM 0-255 selon loi ~log

Note 04

**PwmLedG (v8)**; **PwmLedD (v8)**; **PwmRouge (v8)**; **PwmVert (v8)**; **PwmBleu (v8)**;

## Poussoirs

**PousD** **PousG** valeurs booléennes - **PousD** vaut HIGH si pressé (**PousD !=0**)

**!PousD** **!PousG** **!PousD** vaut HIGH si relâché

Note 05

**nb=GetPous()** ; rend une valeur de 0 à 9 selon le nombre de pressions pendant que cela clignote sur la LedG. **PousG** court augmente; **PousG** 1 seconde diminue. **PousD** ou non action pendant 2 secondes quitte.

## Potentiomètres

Note 06

**GetPotG()**; **GetPotD()**; rendent une valeur 8 bits. Exemple: *byte potG*; *potG= GetPotG()*;  
**GetTemp()**; rend une valeur 10 bits

## Haut-parleur (dessous)

**HpOn**; **HpOff**; **HpToggle**;

La fonction Arduino convient très bien, mais l'intensité dépend de la fréquence  
`tone (14, frequency, duration);`

Note 07

## 7-segment display

**Seg (v8)** ; Allume les segments et le point décimal. Permet toutes les combinaisons.  
**Digit (v4)** ; Allume le chiffre BCD qui correspond au nibble v4 (v4<10)  
**Leds (0xff)** ; **HpOff** ; Segments off et 8 leds vertes off (low power mode).

## Port Ki and leds

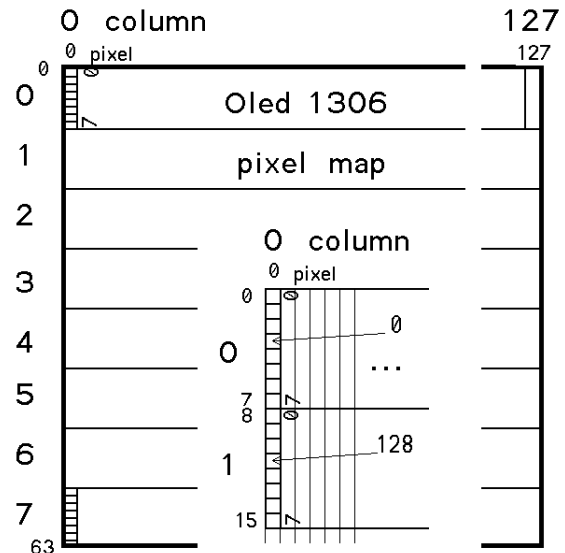
**WrKi (v8)** ; Ecrit un mot de 8 bits sur le port Ki et allume les leds si le bit est à 1.  
**RdKi ()** ; Si le connecteur est soudé, on peut lire la valeur imposée par l'élément connecté.

## Oled.h (OledI2Cbb et OledPix)

```
Car(cc); Text(); Sprite();
Bin8(); Hex8(); Hex16();
Dec8(); Dec9999();
Big(); BigBin8(); BigHex8(); BigHex16();
BigDec8(); BigDec9999();
Dot(x,y); DDot(x,y); Vline(x); Hline(y);
```

```
Sprite(smile); Sprite(bad);
MySprite(toto); byte toto[]={0x3c...
```

```
PosDir(x,y,dx,dy); Ball(x,y);
Raq(x,y,h); Step();
Interaction: variables globales déjà déclarées:
byte x,y,dx,dy,touche,vitRaq
```



Analyse des fonctions Oled sous [www.didel.com/educ/Educ-Graphique.pdf](http://www.didel.com/educ/Educ-Graphique.pdf)

### Compatibilité Arduino

|  |  |   |
|--|--|---|
| LedG pin 4<br>LedD pin 5<br>Rouge pin 6<br>Vert pin 1<br>Bleu pin 7<br>Hp pin 14 | PousG pin pin 2<br>PousD pin pin 3<br>Oled SDA pin 18<br>Oled SCK pin 19<br>SDA SCK Compatible<br>I2C Wire Oled 0x36 Temp 0x48 | Pot gauche Adc6<br>Pot droite Adc7<br>Non documenté Tell / A1 pin 15<br>Port 8 bits<br>Bits 0 – 5 pins 8 à 13<br>Bits 6, 7 pins 16,17 |
|--|--|---|

ADC6 et ADC7 ne sont pas câblés sur les cartes Arduino. Il sont parfois accessibles en A6 A7.  
 Educ utilise une librairie I2C propre. Ajouter la librairie Wire n'a pas été testé.

### Fonctions Arduino intéressantes

Ces fonctions sont installées dès que le compilateur reconnaît le setup – loop d'Arduino.

```
pinmode(pin,dir); (fait par SetupEduC ou SetupLC)
digitalWrite(pin,niv); digitalRead(pin); analogRead(pin);
```

```
bitSet(var,n); bitClear(var,n);
min(a,b) max(a,b);
constrain(x,a,b);
map(value,fromLow,fromHigh,toLow,toHigh);
randomSeed(valeur); random(max); random(min,max);
```

```
Serial.begin(9600); Serial.print(); Serial.println();
```

```
Avec Interruptions timers 0
delayMicroseconds(); pulseIn(pin,b);
analogWrite(pin); pin 5,6
tone(pin,fréqHz); tone(pin,fréqHz,durée); notone();
```