

## Edu-C – Programme Demo avec 15 démos

### Démo

Le programme `Demo.ino` se trouve sous [www.didel.com/educ/Demo.zip](http://www.didel.com/educ/Demo.zip). L'action sur un poussoir sélectionne les 16 programmes possibles, faciles à comprendre, modifier, compléter.

Avec Firefox, il faut extraire le fichier vers le bureau (ne pas ouvrir), puis dans le dossier Demo créé, cliquer sur Demo.ino pour ouvrir avec Arduino. Les bibliothèques sont incluses, n'importe quel PC/Mac avec Arduino installé doit convenir.

Au démarrage le programme affiche un 0. Tant que la Led gauche clignote, on peut agir sur le poussoir gauche pour augmenter la valeur (sature à F). Une pression longue revient en arrière d'un cran. Si on presse sur le poussoir droite, ou sans action pendant quelques secondes, la démo correspondant au no est appelée. Dans plusieurs démos, pour éviter une répétition lassante ou modifier des paramètres, les poussoirs et potentiomètres sont actifs.

Le poussoir "reset" redémarre le choix. Astuce: le dernier choix est mémorisé en EEPROM. Si le poussoir droite est pressé avec le poussoir reset (et maintenu jusqu'à la fin du reset Arduino qui peut inclure la recherche du driver), cet ancien no de test s'exécute immédiatement.

Les démos ont été complétées par des textes qui ne sont pas explicités ci-dessous.

### Démo 0

Les poussoirs agissent sur les leds  
Les deux poussoirs en même temps comptent en binaire.  
Le code apparaît sur les leds vertes, poids faibles en haut, led allumée pour l'état 1. Les segments sont allumés par un état 0.

```
case 0:
    . . .
    while(1){
        if(PousG) {LedGOn;LedDOff;}
        if(PousD) {LedDOn;LedGOff;}
        if(PousG&&PousD) {
            LedGOff; LedDOff; BlancToggle;
            LiCol(4,24); BigBin8 (cn); Leds(cn++);
            DelMs(300); // ralentir le comptage
        }
    }
```

### Démo 1

Les poussoirs agissent sur la led tricolore.  
La valeur numérique du potentiomètre droit est affichée sur les leds vertes.  
Pour le potentiomètre gauche, deux valeurs limites sont détectées pour allumer les leds blanches

```
case 1:
    . . .
    while(1) {
        if(PousG) {RougeOn;}if(!PousG) {RougeOff;}
        if(PousD) {BleuOn;}if(!PousD) {BleuOff;}
        potG = GetPotG(); potD = GetPotD();
        LiCol(4,25); BigHex8 (potG); BigHex8 (potD);
        LiCol(7,30); BigDec8 (potG); BigDec8 (potD);
        if (potG>0x40) {LedGOn;} else {LedGOff;}
        if (potG> 117) {LedDOn;} else {LedDOff;}
        Leds (~potD);
        DelMs (20);
    }
```

### Démo 2

Lit et affiche les potentiomètres.  
Commande l'intensité du Rouge et Vert selon la valeur des potentiomètres.  
Presser sur le poussoir gauche en agitant le EduC.  
L'affichage s'éteint pour que l'on voie mieux les traces de clignotement.  
Les impulsions d'allumage sont faites par interruption, en utilisant des compteurs internes du processeur.

```
case 2:
    LiCol(0,0); Sprite (smile); LiCol(0,100); Sprite (sad);
    LiCol(2,20); Text("Sortie analogique");
    Leds (0xff);
    SetupInter();
    while(1) {
        potG = GetPotG(); potD = GetPotD();
        Rouge (potG/8); Vert (potD/8);
        if (!PousG) {
            LiCol(6,25); BigDec8 (potG);
            LiCol(6,65); BigDec8 (potD);
            DelMs (100);
        }
        else {Clear();}
    }
```

### Démo 3

Lit et affiche les potentiomètres.  
Commande l'intensité du Bleu et Vert selon la valeur des potentiomètres.  
Dessine le diagramme des temps avec l'origine des coordonnées en bas. Il y a 64 pixels verticalement et 128 horizontalement.

```
case 3:
    while(1) {
        potG = GetAnaG(); potD = GetAnaD();
        LiCol(0,25); Dec8 (potG);
        LiCol(0,65); Dec8 (potD);
        Rouge (potG/8); Vert (potD/8);
        Dot (x,63-potG/4); DDot (x,63-potD/4);
        if (x++>=128) {x=0; Clear(); }
        DelMs (20);
    }
```

### Démo 4

La lecture des deux potentiomètres positionne un point en x-y sur l'écran et laisse une trace.  
Le poussoir gauche efface.  
Verticalement, il y a des problèmes à cause de l'écriture qui se fait par segments de 8 bits et on ne peut pas relire l'écran.

```
case 4:
    while(1) {
        potG = GetAnaG(); potD = GetAnaD();
        LiCol(0,20); Dec8 (potG/4);
        LiCol(0,80); Dec8 (potD/4);
        Dot (potG/2,64-potD/4);
        if(PousG) {Clear();}
    }
```

## Démo 5

Test de l'objet Ball qui rebondit sur les côtés. Il faut presser PousG pour permettre le déplacement.  
PousD en même temps génère des balles et perturbe.  
Cette démo est une première étape avant de programmer un ping-pong simple.

```
case 5: . . .
// Hline(0);
Hline(63); Vline(0); Vline(127);
PosDir(64,32,3,4); HpOff;
while(1) { Step(); D;
if (touche==1) dy=-dy;
if (touche==2) dx=-dx;
if (touche==4) dy=-dy;
if (touche==8) dx=-dx;
while (!PousG) ;
if (PousD) {x4++; if (x4>100) x4=20;}
}
```

## Démo 6

Test des raquettes.  
Dans sa définition actuelle, une raquette est verticale et a 3 paramètres; x, y (centre) et longueur.

```
case 6: . . .
byte pot; int pos;
PosDir(40,30,1,4);
while(1) {
pot = GetPotG(); // pos 56 à 8 excursion 48 = 3/16 de 256
pos = 56 - ((3*pot)/16);
Raq(0, pos, 16);
pot = GetPotD(); // pos 56 à 8 excursion 48 = 3/16 de 256
pos = 56 - ((3*pot)/16);
Raq(127, pos, 16);
}
```

## Démo 7

Jeu de réflexe  
Quand la LedG s'allume, il faut presser sur le poussoir G  
Quand la LedD s'allume, il faut presser sur le poussoir D  
Erreur, Led rouge  
Si le temps de réaction est supérieur à 0.5s, Led bleue

```
case 7:
int cnt; byte gd, ok;
#define Temps 300 // temps réaction max
while(1) {
DelMs(1000); //(random(500));
if (random(10)<5) {gd=0; LedGOn;} else {gd=1; LedDOn;}
cnt = 0;
while (cnt++ < Temps) {
DelMs(10);
if (PousG&&(gd==0)) {ok=1; break;}
if (PousG&&(gd==1)) {ok=0; break;}
if (PousD&&(gd==0)) {ok=0; break;}
if (PousD&&(gd==1)) {ok=1; break;}
}
LedGOff; LedDOff;
if (cnt < Temps) { // répondu dans le délai
if (ok==1) {} // continue à jouer
if (ok==0) { RougeOn; DelMs(1000); RougeOff; }
} else { // trop lent
BleuOn; DelMs(1000); BleuOff;
}
} // end while(1)
```

## Démo 8

Musique  
La fonction Arduino tone () est utilisée.  
Le buzzer n'est pas prévu pour jouer des notes; il a des fréquences de résonance très marquées. Il faudrait éviter ces zones.

noTone(14); attend que la note précédente soit terminée. Un délai n'est pas vu autrement.

```
LiCol(2,10); Text("Au clair de la lune");
#define D 500
#define S DelMs(500)
#define FaN S; tone(14,524,D) // noire
#define FaR S; tone(14,524,4*D) // ronde
#define SolN S; tone(14,588,D)
#define SolB S; tone(14,588,2*D) // blanche
#define LaN S; tone(14,660,D)
#define LaB S; tone(14,660,2*D)
while(1) {
BlancOn;
FaN; FaN; FaN; SolN; LaB; SolB;
noTone(14); S; FaN; LaN; SolN; SolN; FaR;
BlancOff;
while (!PousG) ; // on recommence?
}
```

## Démo 9 – Ne fonctionne pas sur les Edu-C "Blonay"

Température  
La valeur est autour de 100. Elle est documentée 0 pour 0 degrés et augmente de 4 par degré avec la fonction v8=GetTemp();

```
while(1) {
temp = GetTemp();
LiCol(7,0); BigHex8(temp); Big('='); Big(' '); BigDec8(temp);
y= (temp)/4; // essayer y= temp-80;
Dot(x,63-y); x++; if(x==128) { Clear(); x=0; }
DelMs(300);
}
```

## Démo 0xA (10) -- Réflexe

On avertit avec un clignotement vert.  
Il y a ensuite un délai aléatoire et il faut agir sur PousD quand la Led s'allume.  
On mesure le temps avec une limite à 250ms pour compter les mauvais réflexes.

```
while(1) {
DelMs(1000); VertOn; DelMs(100); VertOff;
DelMs(random(300,2000));
LedGOn; cont=0;
while (!PousG) { DelMs(10); cont++;
if (cont>25) {ratt++; LedDOn; DelMs(200); LedDOff; cont=0; break;}
}
LedGOff;
LiCol(7,0); BigDec8(cont*10); LiCol(7,90);BigDec8(ratt);
}
```

## . Démo 0xb (11) -- Tirage au hasard

(dans educ/Fun.pfd)

Le programme montre comment déclarer une table de 128 bytes (0 à 127), des compteurs. Quand on tire un nombre entre 0 et 127, on augmente son compteur et on l'affiche graphiquement.

```
//byte taHisto[128]; a définir avant loop();
byte mesure,yy;
//randomSeed(GetPotG());
while(1) {
    mesure = random (0,127+1);
    yy = ++taHisto[mesure];
    Dot (mesure,63-yy);
    if (yy==63) {while(1);} //fini
}
```

## Démo 0xc (12) -- Le montagnard

(dans educ/Fun.pfd)

On dessine une montagne. Avec le pot de gauche, il faut suivre le profil.

```
#define Gpot (GetPotG()/4)
int dl; // on augmentera la vitesse, a 5 on recommence
dl = 50;
#define Del DelMs(dl); LiCol(1,0);BigDec16(tot);
int tot; // total des mauvais points
#define Repete(x) for(byte i=0;i<x;i++)
#define Stop() while(1);
#define Attend(x) while(!x);
while(1) {
    Clear(); x=0; y=0;
    Repete (10) {Dot (x++,63-0);} // plateau
    Repete (40) {Dot (x++,63-y++);} // 45 degrés
    Repete (10) {Dot (x++,63-y);} // plateau
    Repete (20) {Dot (x++,63-y); if(x%2){y--;} } // petite descente
    Repete (5) {Dot (x++,63-y);} // plateau
    Repete (10) {Dot (x++,63-y); y+=3;} // montée raide
    Repete (33) {Dot (x++,63-y--);} // -45 degrés
    // total 127 (on est parti de zéro)
// Attention départ
Digit(3);DelMs(1000);Digit(2);DelMs(1000);Digit(1);DelMs(1000);Digit(0);
LiCol(3,0);BigDec8(dl);
x=0; y=0; tot=0;
Repete (10){Dot(x++,63-y); DDot(x,63-Gpot); tot += abs((Gpot)-y); Del;}
Repete (40){Dot(x++,63-y++); DDot(x,63-Gpot); tot += abs((Gpot)-y); Del;}
Repete (10){Dot(x++,63-y); DDot(x,63-Gpot); tot += abs((Gpot)-y); Del;}
Repete (20){Dot(x++,63-y); DDot(x,63-Gpot); if(x%2){y--;} tot +=abs((Gpot)-y);Del;}
Repete (5) {Dot(x++,63-y); DDot(x,63-Gpot); tot += abs((Gpot)-y); Del;}
Repete (10){Dot(x++,63-y); DDot(x,63-Gpot); y+=3; tot += abs((Gpot)-y); Del;}
Repete (33){Dot(x++,63-y--); DDot(x,63-Gpot); tot += abs((Gpot)-y); Del;}
dl -=5; if(dl==0) dl=50;
Attend(PousG);
}
```

## Démo 0xd (13) -- libre

```
LiCol(2,20); Text("Pas de demo");
while(1) ;
```

## Démo 0xe (14) -- Le générateur de caractère

Les codes de 0 à 31 sont des commandes ignorées.

```
Clear();
LiCol (0,0); for (byte i= 32; i<48; i++) Car(i); // Espace, signes
LiCol (1,0); for (char i= 48; i<64; i++) Car(i); // 0 1 2 3 ...
LiCol (2,0); for (char i= 64; i<80; i++) Car(i); // @ A B C ...
LiCol (3,0); for (char i= 80; i<96; i++) Car(i); // P Q R S ...
LiCol (4,0); for (char i= 96; i<112; i++) Car(i); // ` a b c ...
LiCol (5,0); for (byte i= 112; i<128; i++) Car(i); // p q r s ...
LiCol (7,80); Sprite(didel);
while (1);
```

## Démo 0xf (15) -- Le ping-pong

A partir de ce programme, on peut créer des variantes toujours plus intéressantes.

```
PosDir(64,32,4,1);
while(1) {
    Step();
    y1 = ((256-GetPotG())*3)/16;
    Raq (0,y1);
    y2 = ((256-GetPotD())*3)/16;
    Raq (127,y2);
    Touche();
    if (touch&1) {dy=-dy;}
    if (touch&2) {goto ko;}
    if (touch&4) {dy=-dy;}
    if (touch&8) {goto ko;}
    if (touch&16) {dx=-dx;}
    if (touch&32) {dx=-dx;}
    //while (!PousG);
}
ko:
LiCol(3,50); BigText("KO");LiCol(5,20); Text("PousD recommence");
while(!PousD);
Clear();
nop;
}
```