www.didel.com/educ/EduC-Demos-En.pdf

Edu-C - Demo program with 15 demos

Demo

The Demo.ino program can be found here: www.didel.com/educ/EduC-Demos-En.zip.
The action on a pusher selects the 16 possible programs, easy to understand, modify, complete.
With Firefox, you have to extract the file to the desktop (do not open), then in the created Demo folder, click on Demo.ino to open with Arduino. The libraries are included, any PC / Mac with Arduino installed must be suitable.
At startup the program displays a 0. As long as the left LED is flashing, you can act on the left pusher to increase the value (saturates F). A long press goes back one notch. If you press on the right pusher, or without action for a few seconds, the demo corresponding to the no is called. In several demos, to avoid an annoying repetition or change parameters, pushbuttons and potentiometers are active.
The push button "reset" restarts the choice. Tip: The last choice is stored in EEPROM. If the right pusher is pressed with the reset pushbutton (and held until the end of the Arduino reset which may include the search for the driver), this old test number is executed immediately.
The demos have been supplemented by texts that are not explained below..

Demo 0

The pushers act on the LEDs
The two pushers at the same time count in binary. The code appears on the green leds, low weight up, led on for state 1. The segments are lit by a state 0.

This demo is preceded by a brief test of all input-output.

```
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
        }
      }
```

Demo 1

The pushers act on the tri-color led.
The numerical value of the right potentiometer is displayed on the green leds.
For the left potentiometer, two limit values are detected to light the white LEDs

```
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);
      }
```

Demo 2

Reads and displays the knobs.
Controls the intensity of Red and Green according to the value of the knobs.
Press the left pusher while shaking the EduC. The display turns off so that you can better see the flashing marks.
The ignition pulses are made by interrupt, using internal counters of the processor.

```
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();}
      }
```

Demo 3

Reads and displays the knobs.
Controls the intensity of Blue and Green according to the value of the knobs.
Draw the timing diagram with the origin of the coordinates at the bottom. There are 64 pixels vertically and 128 horizontally.

```
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);
      }
```

Demo 4

Reading both knobs positions a point in x-y on the screen and leaves a trace.
The left pusher erases.
Vertically, there are problems because of writing in 8-bit segments and you can not re-read the screen.

```
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();}
      }
```

Demo 5

Ball object test that bounces off the sides. PousG must be pressed to allow movement.
PousD at the same time generates bullets and disturbs.
This Demo is a first step before programming a simple ping pong.

```
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;}
      }
```

Demo 6

Snowshoe test.
In its current definition, a racket is vertical and has 3 parameters; x, y (center) and length.

As explained in Fun.doc, if the rackets move quickly, the erasure does not have time to do it.

```
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);
      }
```

Demo 7

Reflex game
When the LedG turns on, press the button G
When the LedD lights up, press the D button
Error, red LED
If the reaction time is greater than 0.5s, blue LED

```
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)
```

Demo 8

Music
The Arduino tone () function is used.
The buzzer is not intended to play notes; it has very marked resonant frequencies. These areas should be avoided.

noTone (14); wait until the previous note is finished. A delay is not seen otherwise.

```
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?
 }
```

Demo 9 – Temperature

The value is around 100. It is documented 0 for 0 degrees and increases by 4 per degree with the function 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);
      }
```

Demo 0xA (10) -- Reflex

It is warned with a green flashing.
Then there is a random delay and you have to act on PousD when the LED comes on. Time is measured with a limit of 250ms to count the bad reflexes.

```
     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);
     }
```

Demo 0xb (11) -- Random draw

(in educ/Fun.pdf)
The program shows how to declare a table of 128 bytes (0 to 127), counters. When you draw a number between 0 and 127, you increase its counter and display it graphically.
A reset to reset the table to zero (a for loop could do this).

```
//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
 }
```

Demo 0xC (12) -- The mountaineer

(in educ/Fun.pdf)

We draw a mountain.
With the left pot, you have to follow the profile..

```
 #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);
}
```

Demo 0xd (13) - free - completed on 14 feb

Move the pointer to the randomly placed ball

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

Demo 0xE (14) -- The character generator

Codes from 0 to 31 are ignored commands.

```
       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);
```

Demo 0xF (15) -- Ping pong

(in educ/Fun.pdf)

From this program, we can create ever more interesting variants.

A return counter has been added. We count in touchdowns 16 and 32, and we display the score before starting again.

```
      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;
}
```