

Composition du shield

1. Boutons

Le shield comporte 4 boutons, dont 2 possèdent une interruption spécifique comme indiqué dans le tableau suivant :

Bouton	Position	Arduino Pin	Port	Interruption	Résistance de tirage
bp0	Bas Gauche	2	PD2	0	Pull Up
bp1	Haut Gauche	3	PD3	1	Pull Up
bp2	Bas Droite	A0	PC0		Pull Down
bp3	Haut Droite	A1	PC1		Pull Down

Principe d'utilisation

```
const char bpPin[4]={2,3,A0,A1};           // Position physique des boutons
char bpEtat[4];
char bpAncien[4];

void setup()
{
    char i;
    for (i=0;i<4;i++) pinMode(bpPin[i],INPUT); // Déclaration des 4 entrées
}

void loop()
{
    char i;
    for (i=0;i<4;i++)
    {
        bpAncien[i]=bpEtat[i];           // Conservation des états précédents
        bpEtat[i]=digitalRead(bpPin[i]); // Lecture des nouveaux états
    }
    if ((bpEtat[0]==0)&&(bpAncien[0]==1)) // En cas d'appui sur bp0
    {
        .....
    }
    .....
}
```

2. Leds

6 leds connectées en "cathodes communes" sont présentes sur la carte. Attention, le commutateur permet de choisir entre les Leds et les afficheurs 7 segments. Leur position et couleur sont données dans le tableau suivant

Numéro	f5	f4	f3	f2	f1	f0	p1	p0
Couleur	r	o	v	r	o	v	v	r
Arduino Pin	13	12	11	10	9	8	7	6
Port	PB5	PB4	PB3	PB2	PB1	PB0	PD7	PD6

Principe d'utilisation

```

void setup()
{
    char i;
    for (i=6;i<=13;i++) pinMode(i,OUTPUT);    // Déclaration des 6 sorties
}

void loop()
{
    char i;
    for (i=6;i<=13;i++) digitalWrite(i,HIGH); // Les Leds s'allument
    delay(500);                               // Attente pendant 500ms
    for (i=6;i<=13;i++) digitalWrite(i,LOW);  // Les Leds s'éteignent
    delay(500);                               // Attente pendant 500ms
}

```

3. Afficheurs 7 segments

/wikigeii/index.php/Fichier:7seg.png[]

Les 2 afficheurs ne peuvent pas être utilisés simultanément. L'état de la sortie mux (arduino port 4 ou PD4) permet de sélectionner l'un ou l'autre. En allumant successivement l'un puis l'autre rapidement, on a l'illusion qu'ils sont tous 2 allumés. Les segments des afficheurs sont câblés de façon analogue comme décrit ci dessous :

Segment	a	b	c	d	e	f	g	pt
Arduino Pin	13	12	6	7	8	10	9	11
Port	PB5	PB4	PD6	PD7	PB0	PB2	PB1	PB3

Principe d'utilisation

```

const char pinMux = 4;
const char pinAff[8]={13,12,6,7,8,10,9,11};

void setup()
{
    char i;

    for (i=0;i<8;i++) pinMode(pinAff[i],OUTPUT);    // Déclaration des 8 sorties des
    pinMode(pinMux,OUTPUT);                        // + sortie de multiplexage (cho

}

void loop()
{
    char i,c;

    for (i=0;i<8;i++) digitalWrite(pinAff[i],HIGH); // Les segments s'allument
    for (c=0;c<20;c++)
    {
        digitalWrite(pinMux,1);                    // sur l'afficheur 1
        delay(10);
        digitalWrite(pinMux,0);                    // puis sur l'afficheur 2
        delay(10);
    }

    for (i=0;i<8;i++) digitalWrite(pinAff[i],LOW); // Les segments s'éteignent
    for (c=0;c<20;c++)
    {
        digitalWrite(pinMux,1);                    // sur l'afficheur 1
        delay(10);
        digitalWrite(pinMux,0);                    // puis sur l'afficheur 2
        delay(10);
    }
}

```

```
}  
  
}
```

4. MLI (ou PWM)

Une sortie MLI non filtrée ou après un filtre passe bas de fréquence de coupure 50Hz est disponible sur la carte. Cette sortie est reliée sur l'arduino Pin 5 ou PD5.

Principe d'utilisation

```
const unsigned char mliPin=5;  
  
void setup()  
{  
  pinMode(mliPin, OUTPUT);  
}  
  
void loop()  
{  
  analogWrite(mliPin, valeur);    // valeur de mli entre 0 et 255  
}
```

5. Entrées analogiques

Capteur	Arduino Pin	Port
photo coupleur par réflexion	A2	PC2
LDR, intensité lumineuse	A3	PC3
CTN, thermistance	A4	PC4
mesure de composant (*)	A5	PC5

(*) Mesure de composant : en plaçant un composant sur les barrettes adéquates, et en utilisant l'entrée analogique A5 ou PC5, il est possible d'identifier le composant inséré, la mesure effectuée étant un pont diviseur entre ce composant la sortie 5 et une résistance de 10 kΩ.

Principe d'utilisation

```
void setup()  
{  
  Serial.begin(9600);  
}  
  
void loop()  
{  
  Serial.println(analogRead(A3), DEC);  
  delay(500);  
}
```