

# Parcours autour de Pong !

[une progression pour découvrir Arduino]

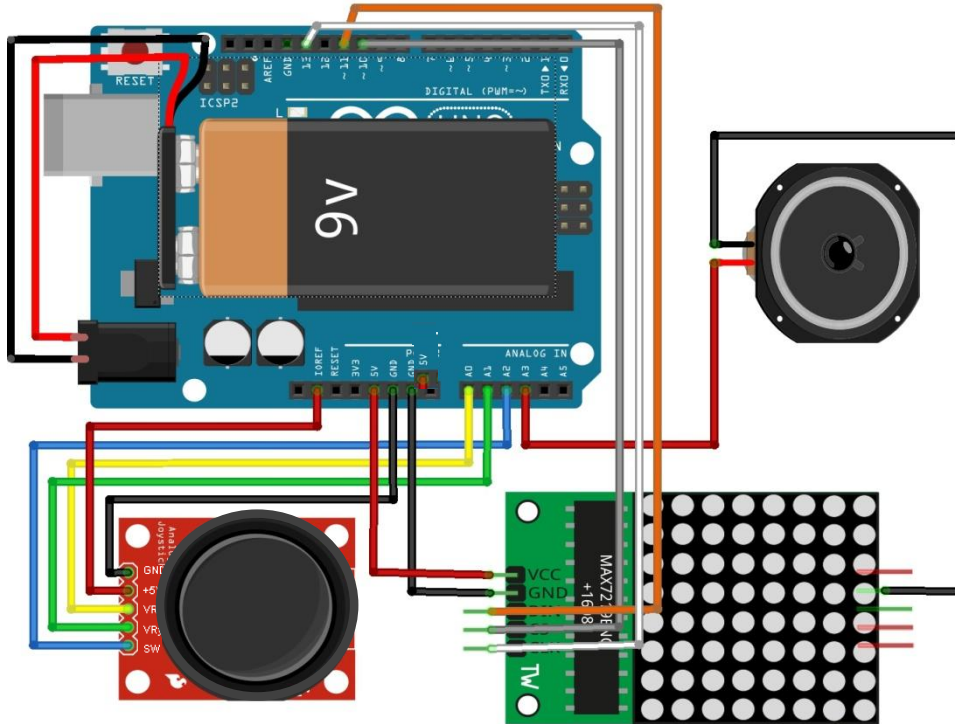
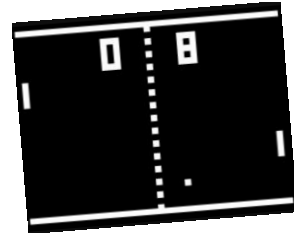
*Vous avez des idées ?  
Venez les fabriquer !*



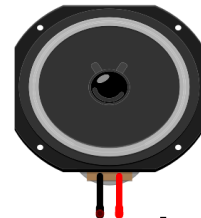
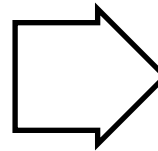
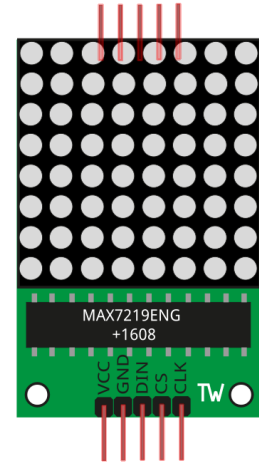
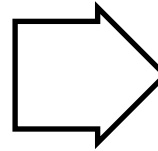
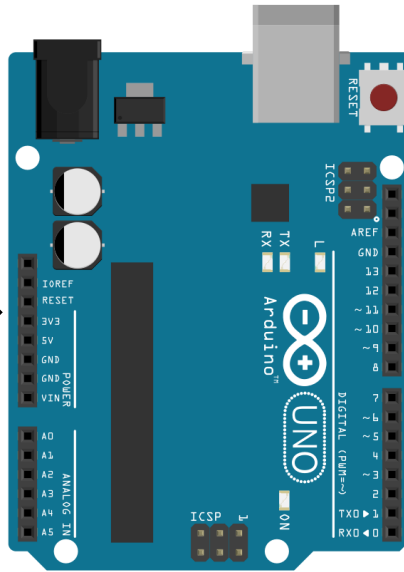
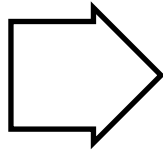
*Parthi* **LAB**

Un jeu d'arcade vintage?

# PONG !



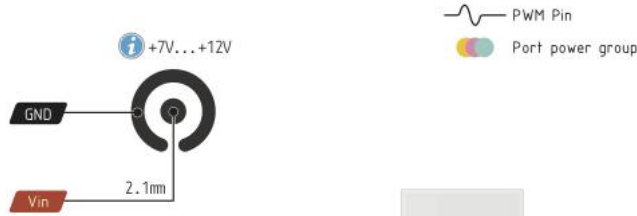
# Arduino



informations en  
entrée

programme de  
traitement

commande en  
sortie

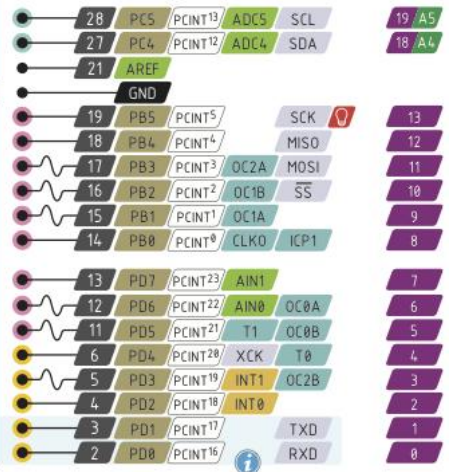
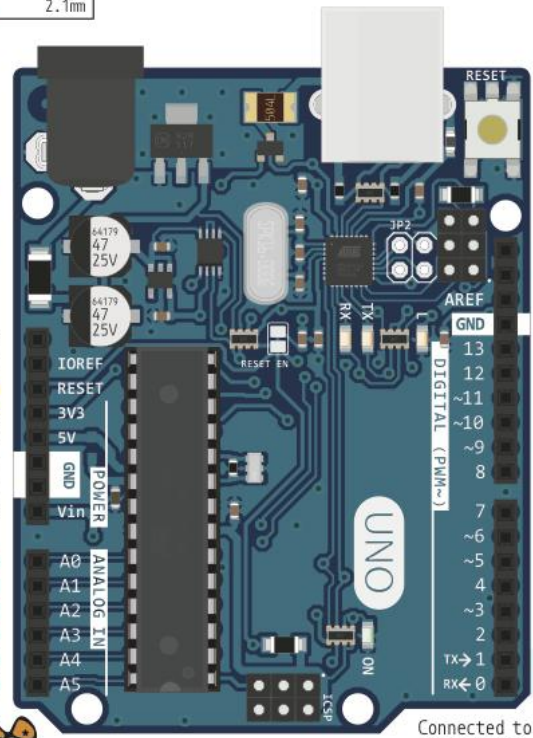
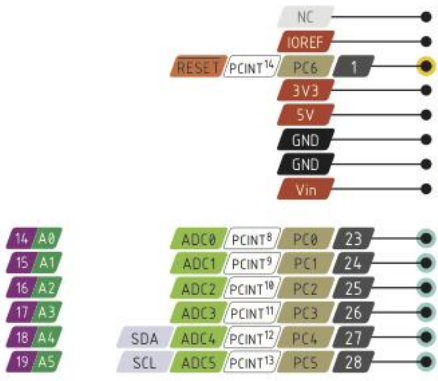


Power	Serial PIN
GND	PIN Function
Physical PIN	Interrupt PIN
Port PIN	Control PIN
Analog PIN	IDE

Input voltage to the board when it's using an external power source. Not USB bus power!

Logic reference voltage for shields Connected to the 5V bus

- Absolute** MAX per pin 40mA, 20mA recommended
- Absolute** MAX 200mA for the entire package
- The total current of each port power group **should not exceed** 100mA



Connected to the ATmega and used for USB programming and communicating with it

produire un son

tone()



```
#define HP 17 //+HP sur A3  
int frequence = 196;  
int duree = 200;  
int pause = 2000;
```

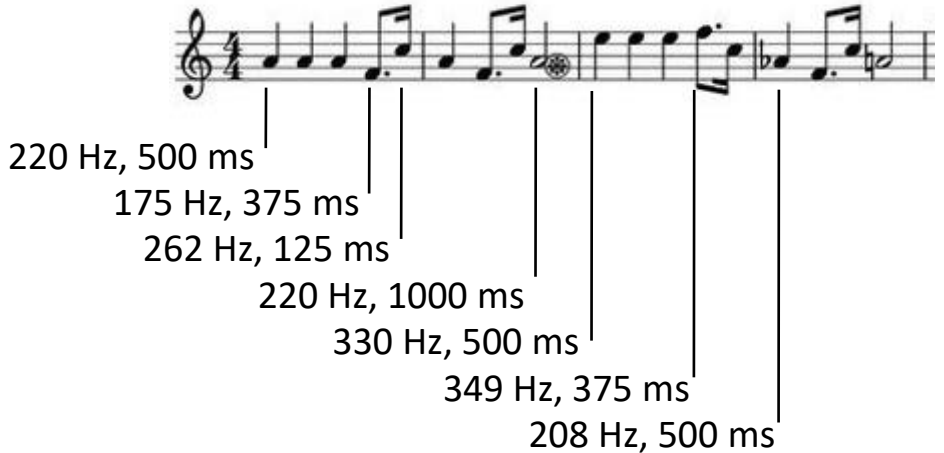
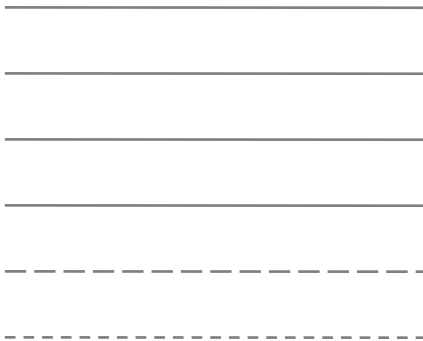
```
void setup() {  
  pinMode( HP, OUTPUT);  
}
```

```
void loop() {  
  tone(HP, frequence, duree);  
  delay(pause);  
}
```

# jouer la marche impériale



```
tone(HP,220,500); delay(501);  
tone(HP,220,500); delay(501);  
tone(HP,220,500); delay(501);  
tone(HP,175,375); delay(376);  
tone(HP,262,125); delay(126);
```



# jouer la marche impériale

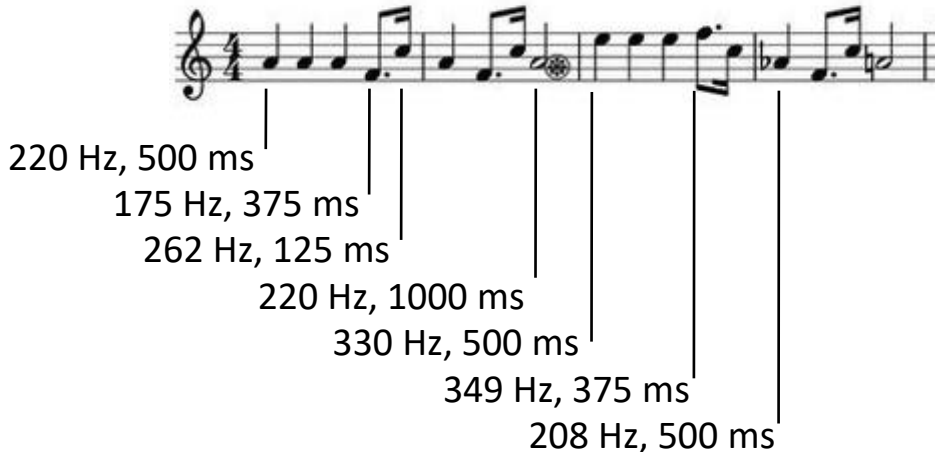


tone(HP,220,500); delay(501);  
tone(HP,220,500); delay(501);  
tone(HP,220,500); delay(501);  
tone(HP,175,375); delay(376);  
tone(HP,262,125); delay(126);

tone(HP,220,500); delay(501);  
tone(HP,175,375); delay(376);  
tone(HP,262,125); delay(126);  
tone(HP,220,1000); delay(1001);

tone(HP,330,500); delay(501);  
tone(HP,330,500); delay(501);  
tone(HP,330,500); delay(501);  
tone(HP,349,375); delay(376);  
tone(HP,262,125); delay(126);

tone(HP,208,500); delay(501);  
tone(HP,175,375); delay(376);  
tone(HP,262,125); delay(126);  
tone(HP,220,1000); delay(1001);



# déclencher un événement

# digitalRead()



```
#define HP 17 //+HP sur A3
#define bouton 16 //BP sur A2
boolean boutonState;

void setup() {
  pinMode(HP, OUTPUT);
  pinMode(bouton, INPUT_PULLUP);
}

void loop() {
  boutonState = digitalRead(bouton);
  if (boutonState == HIGH) {noTone(HP);}
  else {tone(HP,330,100);}
}
```



# lire une valeur analogRead()



```
#define potar A0
int valeur = 0;

void setup() {
  pinMode(potar,INPUT);
  Serial.begin(9600);
}

void loop() {
  valeur = analogRead(potar);
  Serial.println(valeur);
}
```

# contrôler un son

# analogRead()



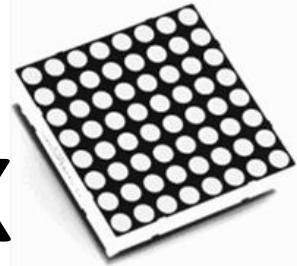
```
#define potar A0
#define HP 17 //+HP sur A3
int valeur = 0;

void setup() {
  pinMode(potar,INPUT);
  pinMode(HP, OUTPUT);
}

void loop() {
  valeur = analogRead(potar);
  if (valeur < 100 || valeur > 1000)
  {tone(HP,valeur);}
  else {noTone(HP);}
}
```

afficher une image

# matrix



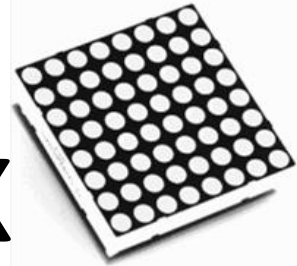
```
#include <SPI.h>
#include <Adafruit_GFX.h>
#include <Max72xxPanel.h>
int pinCS = 10; //CS sur 10, DIN sur 11 (MOSI), CLK sur 13 (SCK)
int AffichH = 1; //un seul afficheur à l'horizontale
int AffichV = 1; // ... et à la verticale
//Création de la matrice « matrice"
Max72xxPanel matrice = Max72xxPanel(pinCS, AffichH, AffichV);

void setup() {
matrice.setIntensity(0); //Réglage de l'intensité lumineuse, de 0 à 15
matrice.fillScreen(LOW); //matrice éteinte

void loop() {
matrice.drawPixel( 1,5, HIGH ); //pixel (1,5) allumé
matrice.drawLine( 0,0,7,7, HIGH ); //ligne du point (0,0) au point (7,7)
matrice.write(); //Envoi du dessin sur la matrice
```

afficher une cible

# matrix



```
matrice.drawLine( 0,0,0,7, HIGH );  
matrice.drawLine( 0,7,7,7, HIGH );  
matrice.drawLine( 7,7,7,0, HIGH );
```

---

---

---

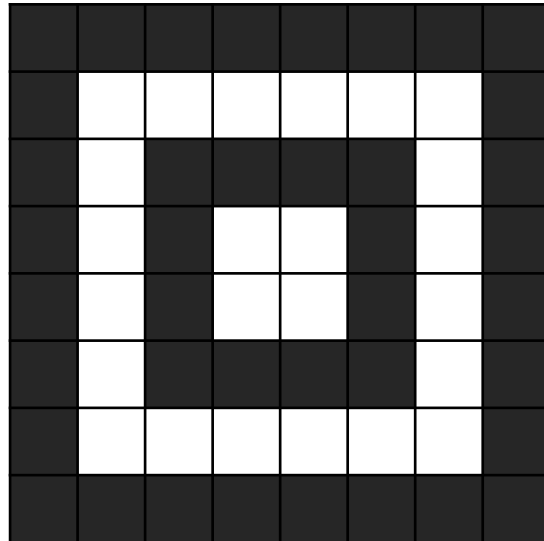
---

---

---

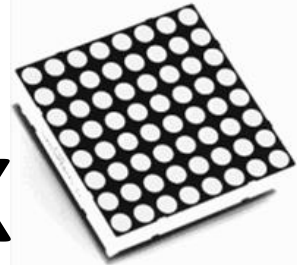
---

---



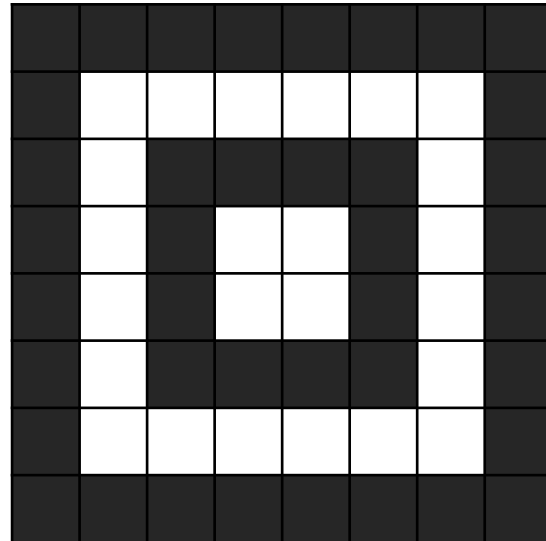
afficher une cible

# matrix



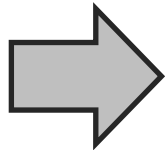
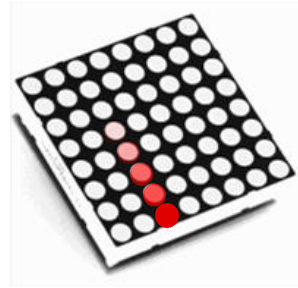
```
matrice.drawLine( 0,0,0,7, HIGH );  
matrice.drawLine( 0,7,7,7, HIGH );  
matrice.drawLine( 7,7,7,0, HIGH );  
matrice.drawLine( 7,0,0,0, HIGH );
```

```
matrice.drawLine( 2,2,2,5, HIGH );  
matrice.drawLine( 2,5,5,5, HIGH );  
matrice.drawLine( 5,5,5,2, HIGH );  
matrice.drawLine( 5,2,2,2, HIGH );
```



créer un protopong

# PiNG!



La position du joystick est représentée par un pixel.  
Un son donne l'alerte lorsque le pixel touche un bord.

La fonction *map()*, pour ré-étalonner une valeur d'une plage vers une autre plage

**map(value, fromLow, fromHigh, toLow, toHigh)**

value: le nombre à ré-étalonner

fromLow: limite inférieure de la plage actuelle de la valeur

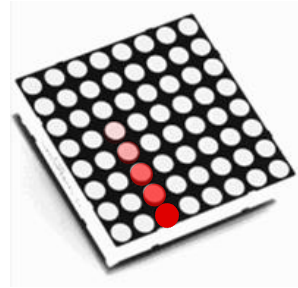
FromHigh: limite supérieure de la plage actuelle de la valeur

toLow: limite inférieure de la plage cible de la valeur

toHigh: limite supérieure de la plage cible de la valeur

# créer un protopong

# PiNG!

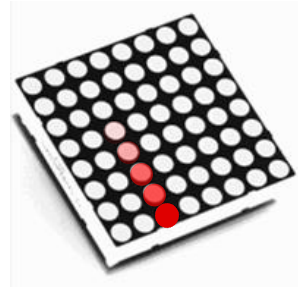


```
#include <SPI.h>
#include <Adafruit_GFX.h>
#include <Max72xxPanel.h>
#define potarH A0
#define potarV A1
#define HP 17
int H = 0; int V = 0; int pinCS = 10;
int AffichH = 1;
int AffichV = 1;
Max72xxPanel matrice = Max72xxPanel(pinCS, AffichH, AffichV);

void setup() {
  matrice.setIntensity(0); matrice.fillScreen(LOW);
  pinMode(potarH,INPUT); pinMode(potarV, INPUT);
  pinMode(HP, OUTPUT);
}
```

créer un protopong

# PiNG!



```
void loop() {  
  H = analogRead(potarH); V = analogRead(potarV);  
  H = map(H,0,1023,0,7); V = map(V,0,1023,0,7);  
  matrice.fillScreen(LOW); matrice.drawPixel( H,V, HIGH );  
  matrice.write();  
  noTone(HP);  
  if (H == 0) {tone(HP,100);}  
  if (V == 0) {tone(HP,200);}  
  if (H == 7) {tone(HP,300);}  
  if (V == 7) {tone(HP,400);}  
  delay(100);}  
}
```



# Parth'Lab

[fablab à Parthenay]

5 rue Jean Macé

79200 Parthenay

05 49 71 08 70

parthlab@cc-parthenay-gatine.fr

*L'expérience n'est pas téléchargeable mais on peut la partager ;)*

[www.parthlab.fr](http://www.parthlab.fr)

[www.facebook.com/parthlab/](https://www.facebook.com/parthlab/)