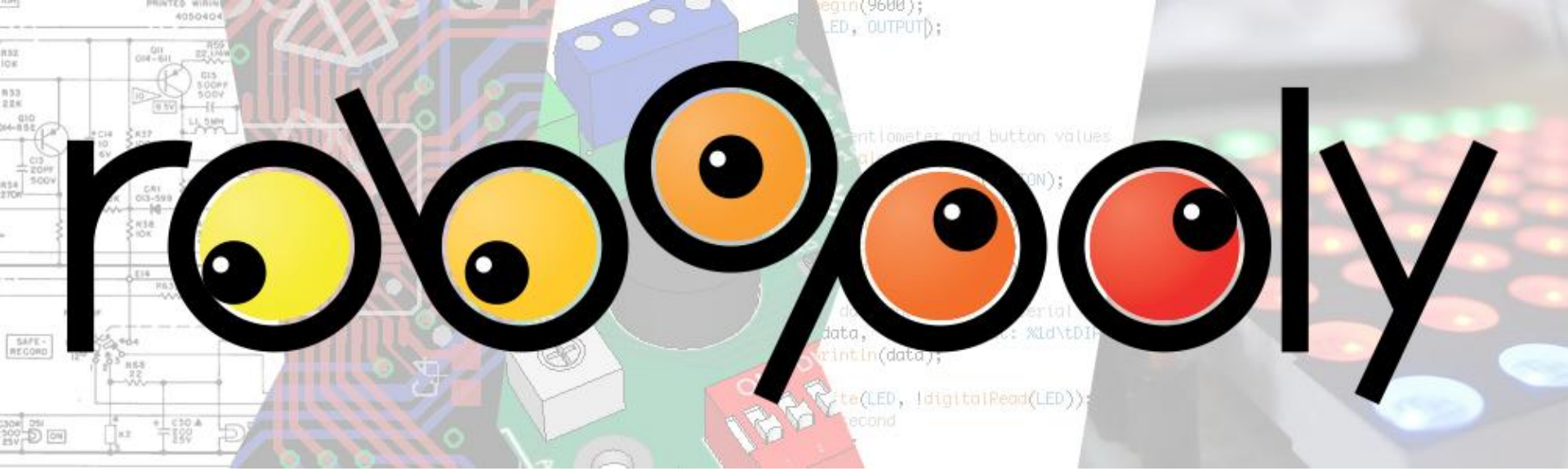


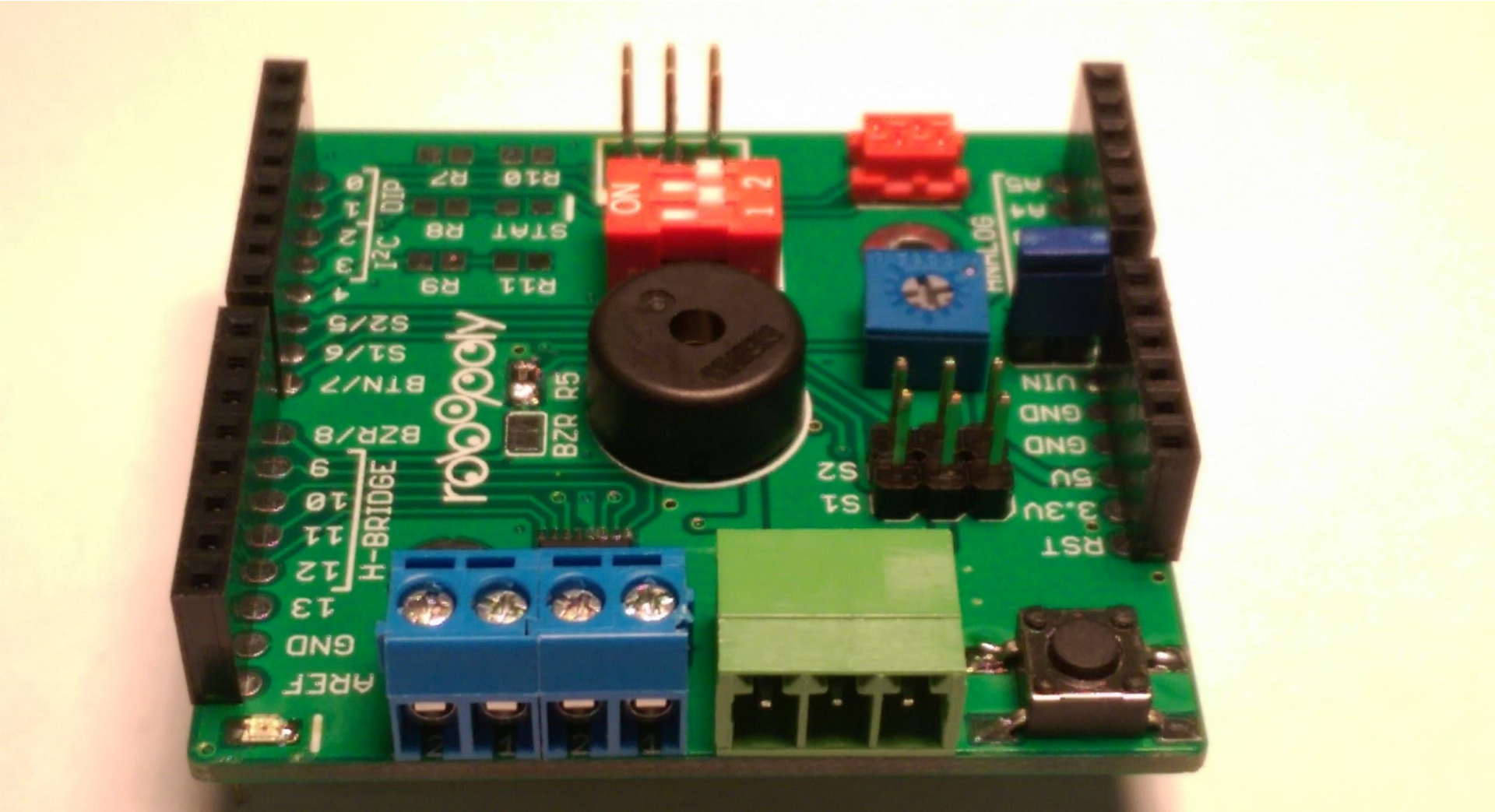
SHIELD

PROGRAMMATION

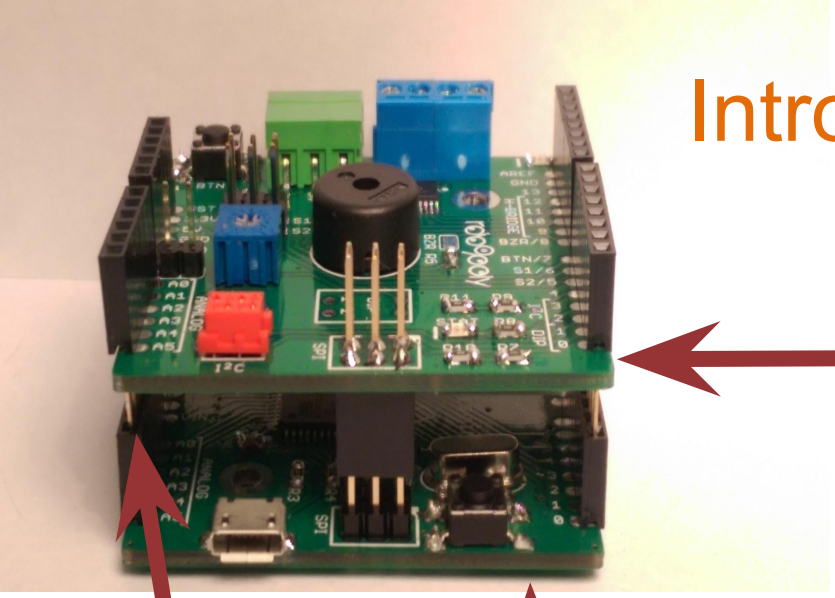
Programmation & Shield

Comment programmer son robot et utiliser les accessoires présents sur le shield

Le Shield


I2C DIP
START
H-BRIDGE
BZR
ON
VIN
GND
GND
5V
3.3V
RST
S1
S2
AREF
GND
13
12
11
10
9
8
7
6
5

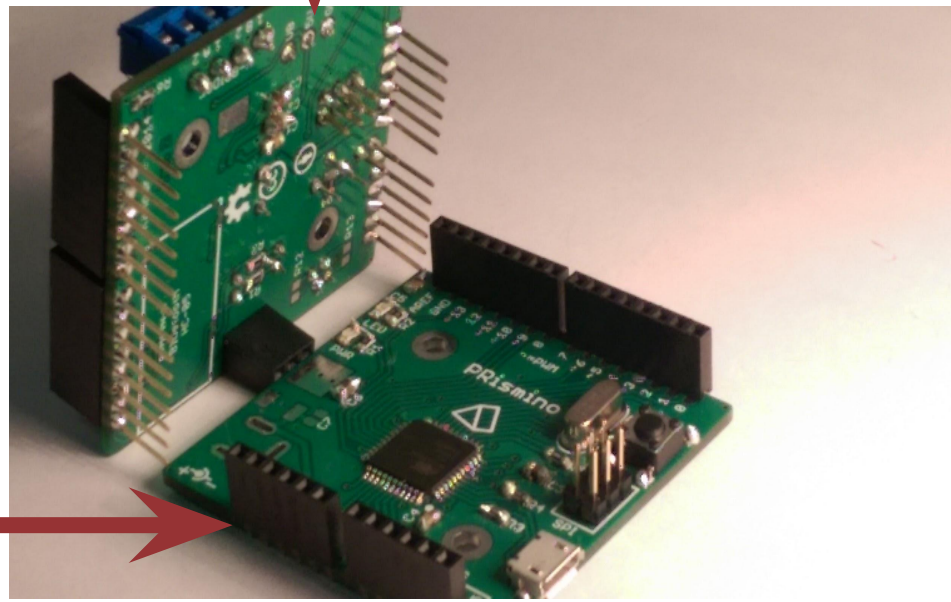
Introduction



Le shield

Les pins

La carte microcontrôleur



- Communication entre les deux cartes
- Emboitement des deux cartes
- compatibilité Arduino Leonardo

Un peu de vocabulaire

Les termes d'alimentation :

- Vcc = Vs = V_{logic} = 5V
- GND = Ground = Mass = 0V
- Vb = $V_{batterie}$ = V_{moteur}

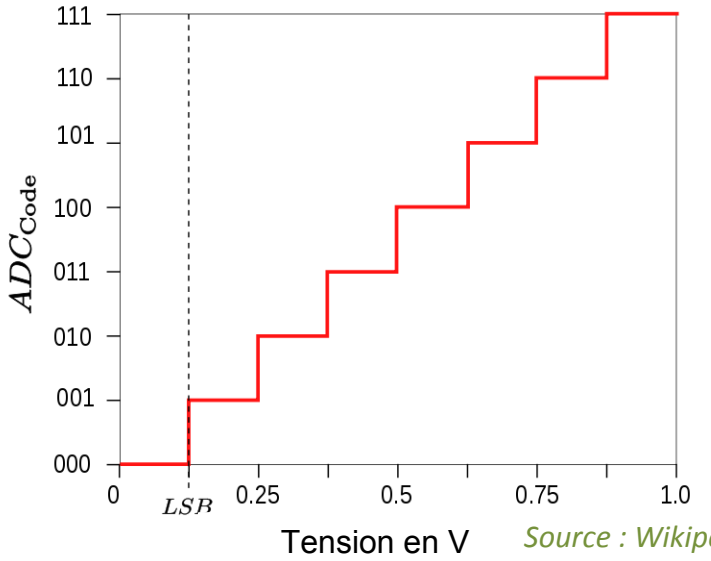
Signal logique :

- Condition Vraie ou Fausse
- Faux = 0V = 0 logique
- Vrai = 5V = 1 logique
- Tout message numérique suite de 0 et 1

Exemple : Interrupteur, LED, etc...

Signal analogique :

- Signal continu dans le temps
- L'information est contenu dans la tension !
- Il existe une relation entre la tension et l' information qu'elle représente



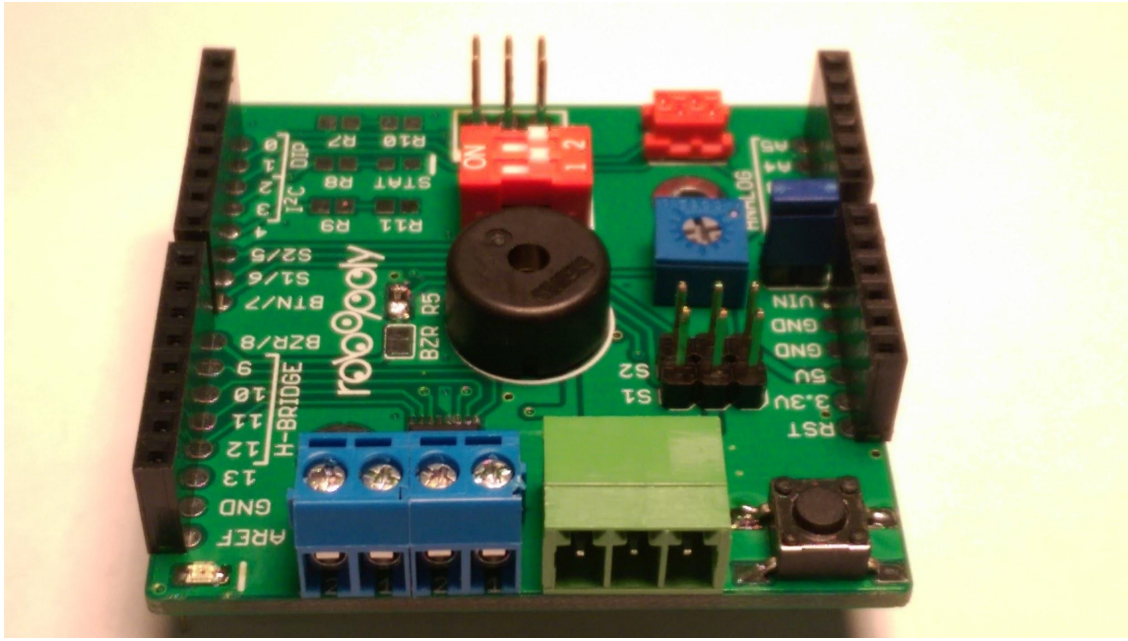
Source : Wikipedia

Exemple : Capteur de distance, d'humidité, de température, etc...

Les éléments intégrés sur le Shield

Les éléments par défaut :

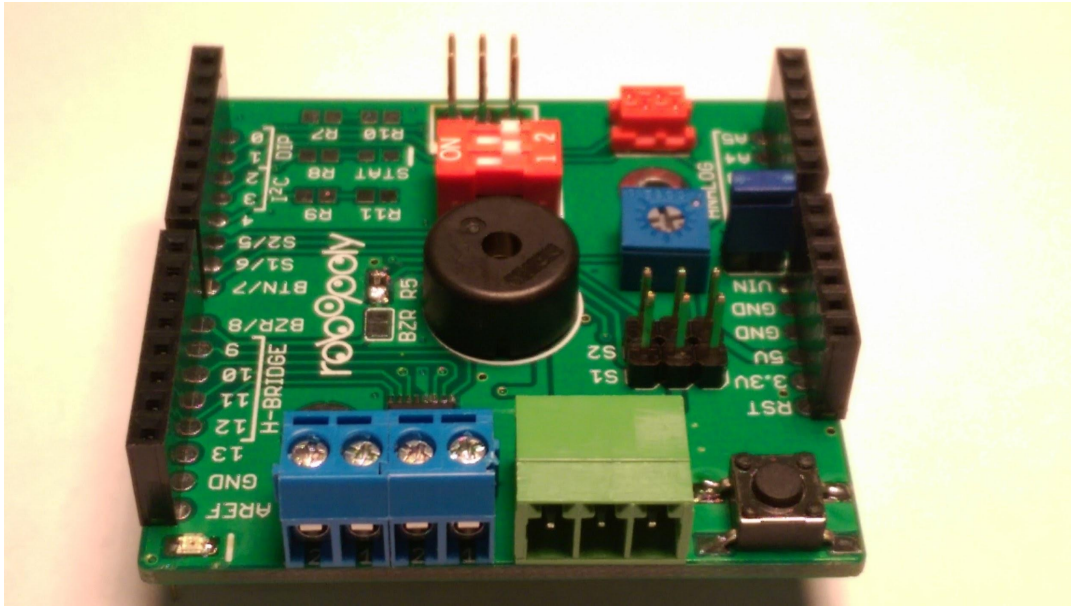
- L'alimentation des moteurs
- Le connecteur d'alimentation
- Le bouton poussoir
- Les connecteurs des servomoteur
- Le Buzzer
- Le potentiomètre
- Le DIP switch



Les éléments intégrés sur le Shield

Les branchements externes :

- Le port de Conversion Analogique Numérique (ADC)
- Les pins de Conversion Numérique Analogique (DAC)
- Les pins d'interruption
- Les pins d'alimentation
- Les pins de PWM
- Les pins logiques entrés/sorties



WARNING : Chaque pin permets plusieurs fonctions, mais pas en même temps

Le Potentiomètre et le buzzer

Potentionmètre : pin A0

Buzzer : pin 8

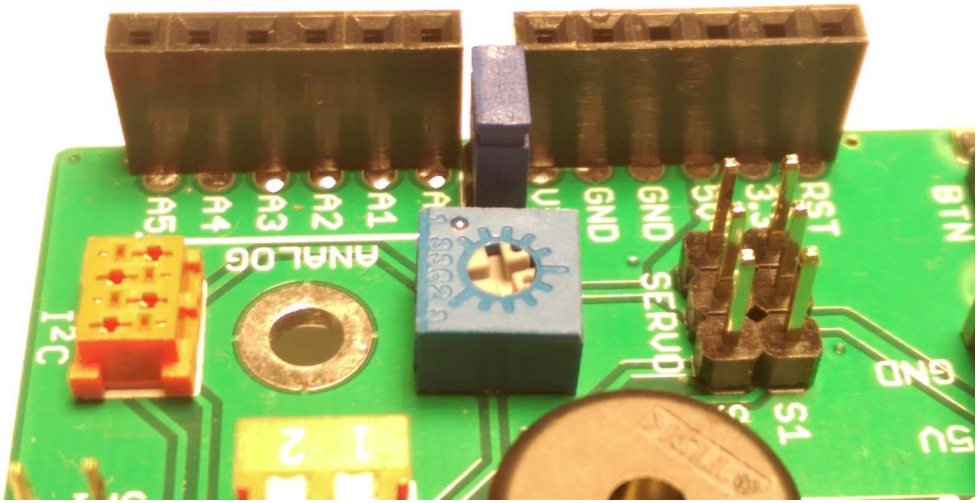
Bouton: pin 7

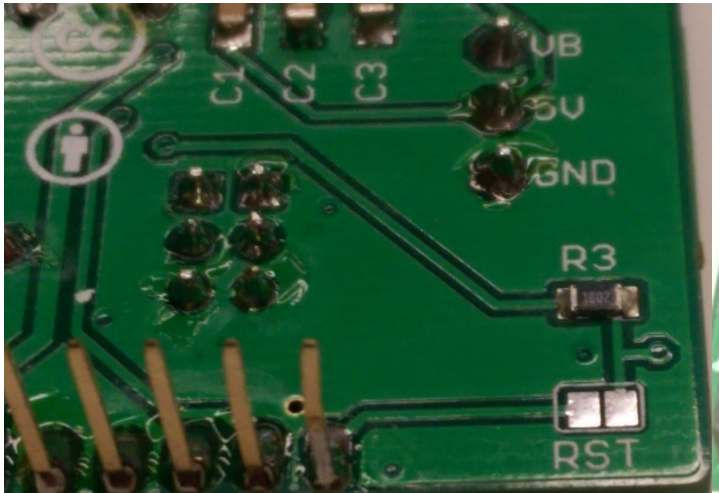
Description et possibilités :

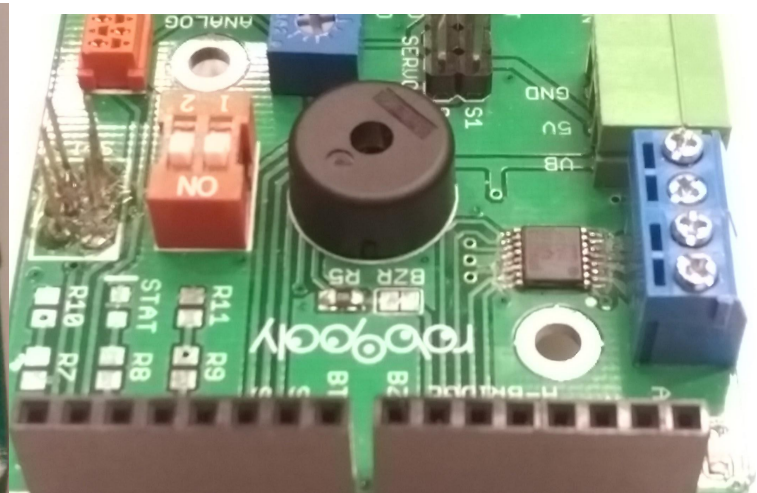
Buzzer:pont soudure, fermé à l'origine

Reset: pont soudure, ouvert

Potentiomètre:jumper



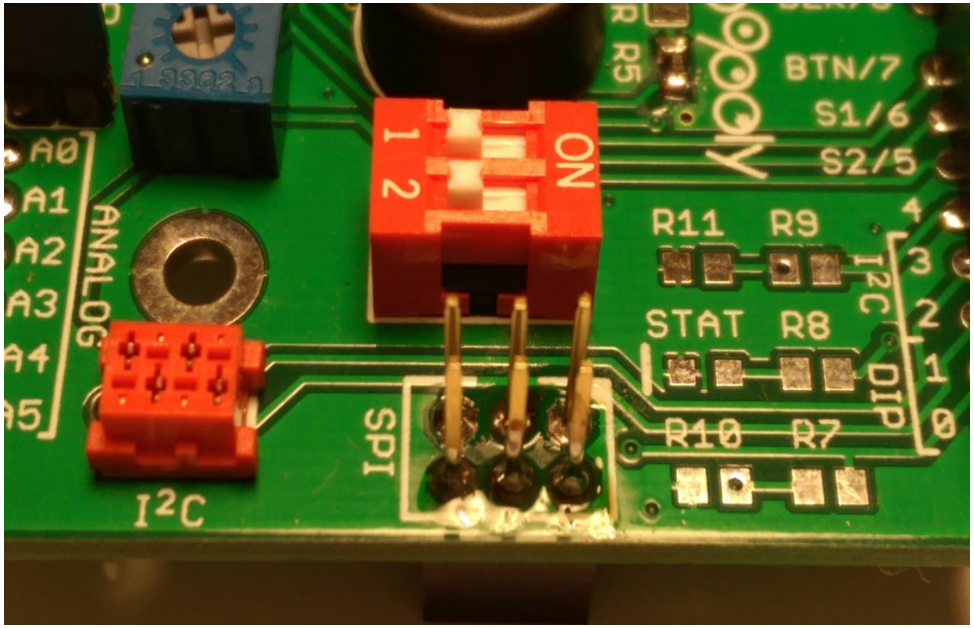




Les éléments intégrés sur le Shield

Les éléments de communications:

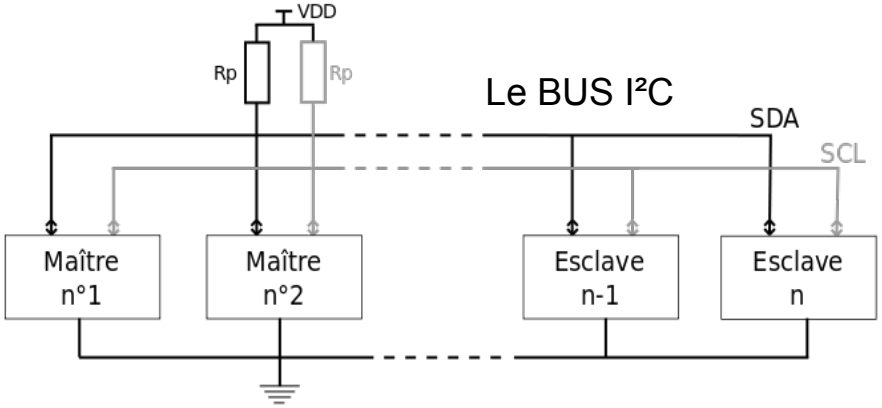
- La liaison Série (UART)
- Le protocole I²C avec son connecteur micromatch
- Le module Bluetooth
- Le protocole SPI



Ces éléments sont les caractéristiques du microcontroleur vous pouvez donc retrouvez tous ces détails et bien plus encore sur :

http://arduino.cc/en/Main/arduinoBoardLeonardo

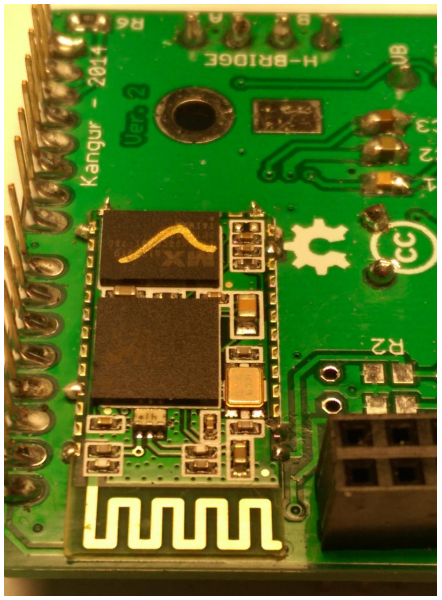
Le protocole I²C



- Communications entre différents modules
  - Mêmme entre plusieurs PRismino
  - Modules provenant des internet ou dévéloppés à robopoly

- Beaucoup de capteurs du marché utilisent ce protocole (Accéléromètre, Gyroscope etc…)

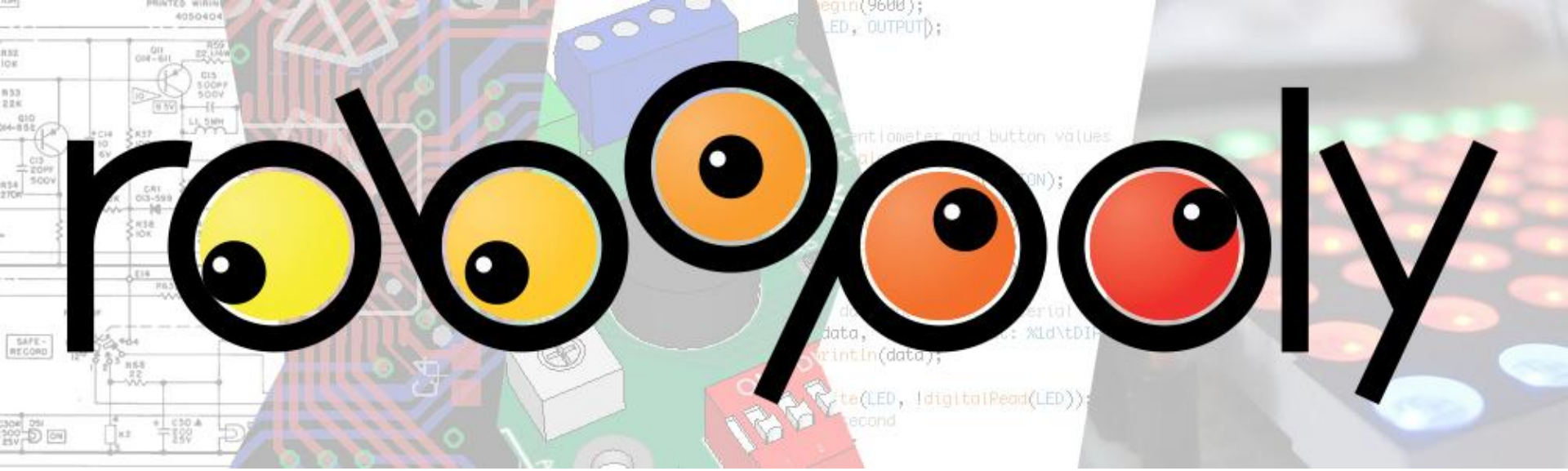
Le module Bluetooth



- Utilisation de la liaison série (pin 0 et 1)
-
- Se contrôle via des Commandes AT
-
- Permet une communication sans fil avec un autre périphérique (Ordinateur, smartphone, autre prisme)

La documentation est disponible sur notre site !

Plus d'informations lors des prochains démons

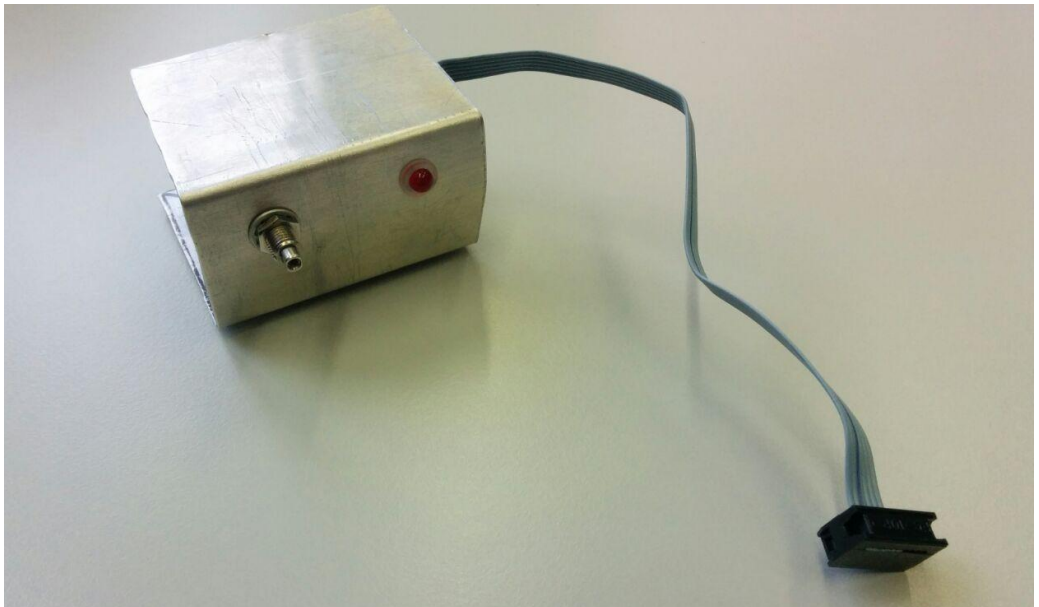


SHIELD

PROGRAMMATION

Le bootloader

- Programme qui s'execute au reset
- Permet d'être detecté par l'ordinateur
- Reprogrammer le microcontrôleur
- Comment l'installer? => Comité/ nouveau boitier



Arduino

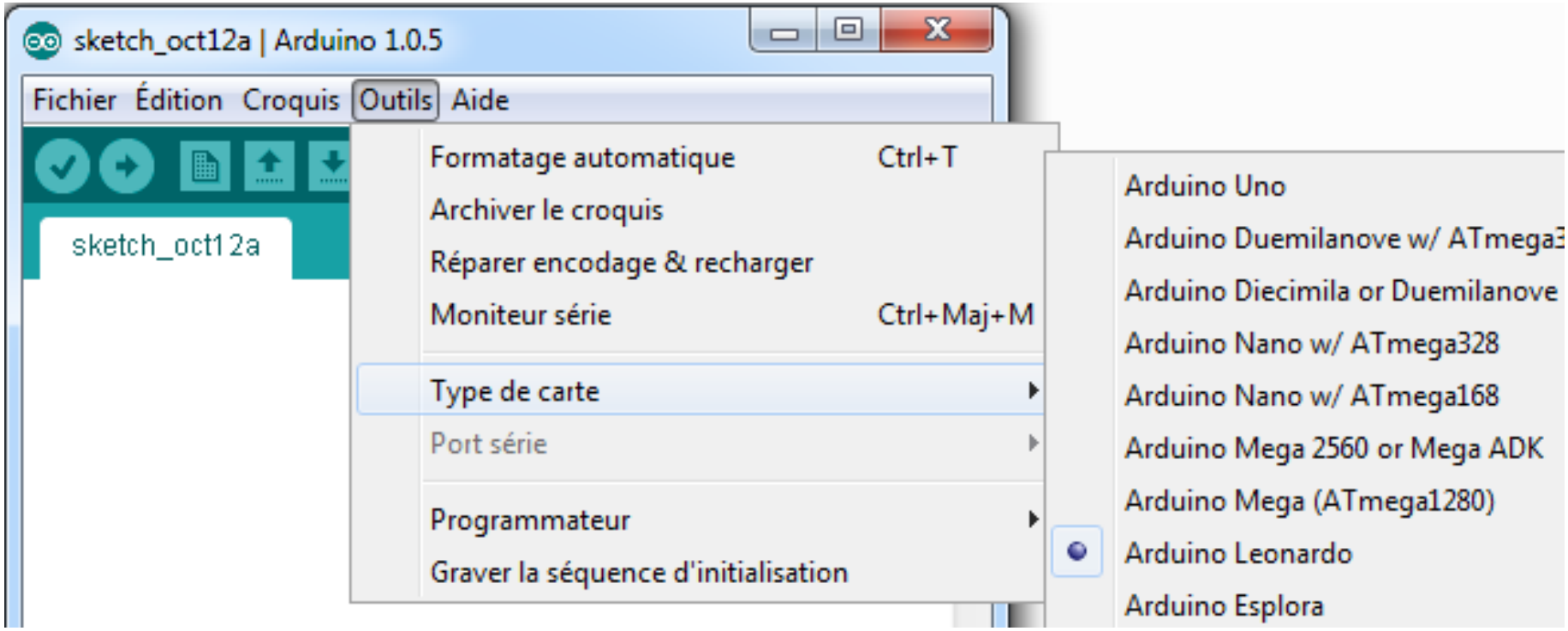
- Installer Arduino, voir robopoly.epfl.ch sous la rubrique Programmation
- Documentation sur le site Arduino: www.arduino.cc/en/Reference

- Possibilité d'utiliser AVRStudio6

  http://robopoly.epfl.ch/cms/site/robopoly/lang/fr/prisme/tutoriels/atmelstudio

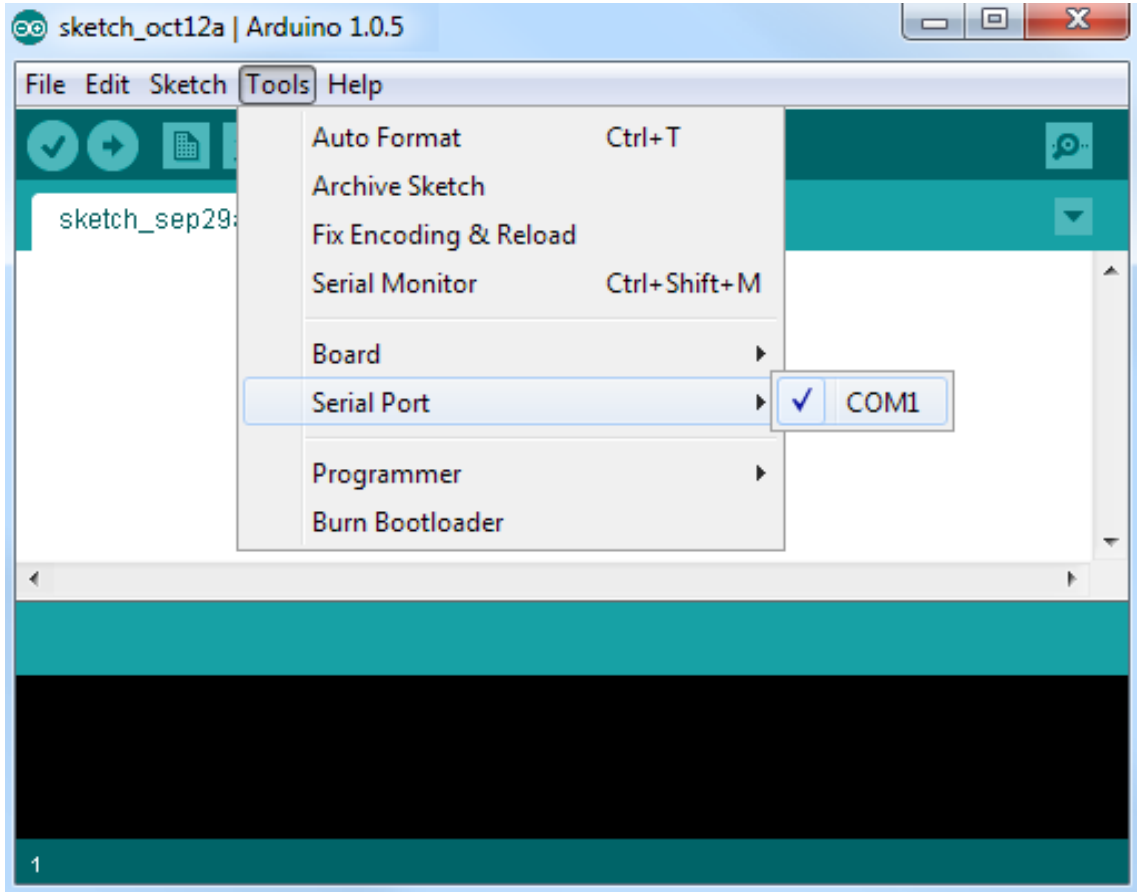
  et autres IDE

Configuration d'Arduino



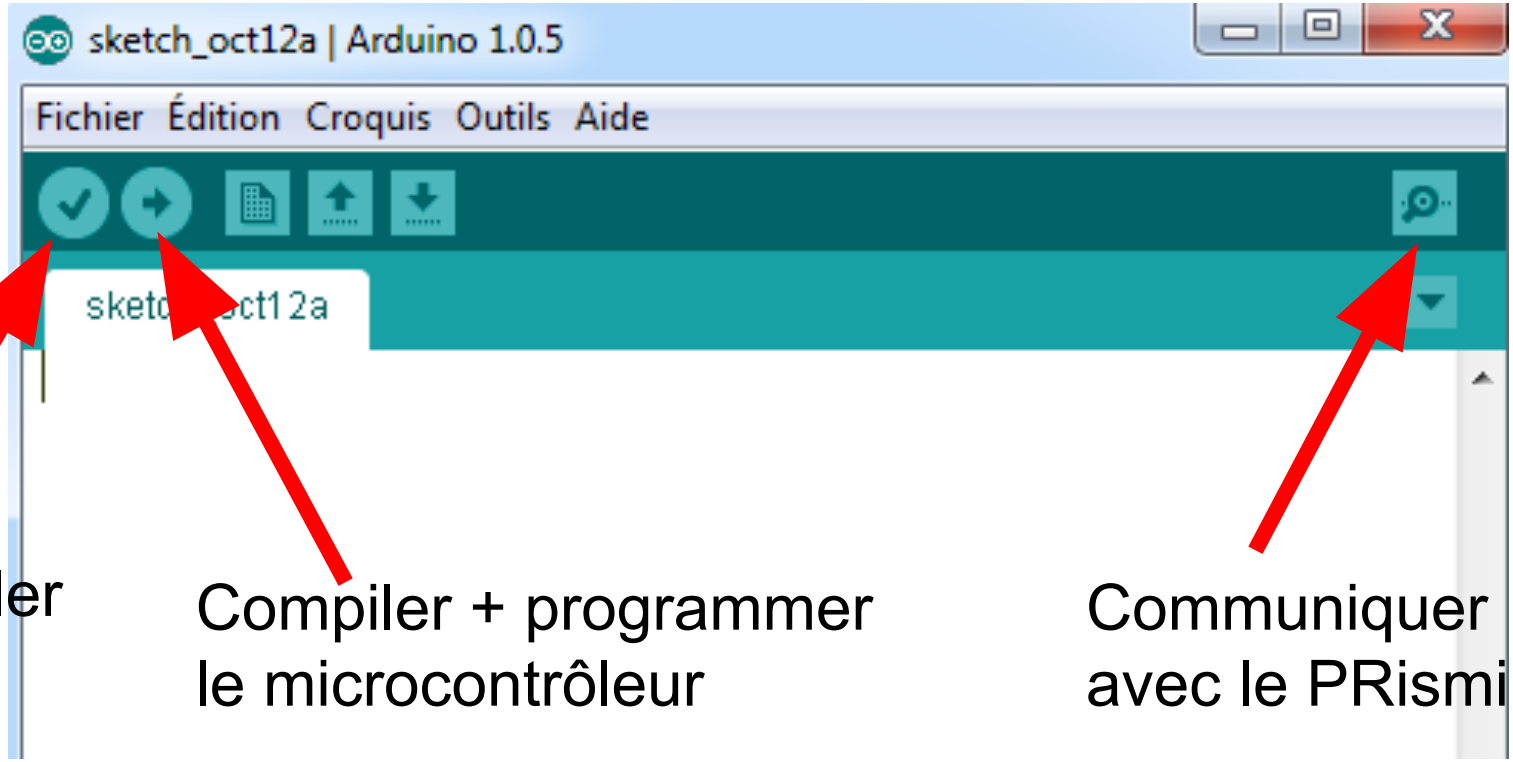
Type de carte: Arduino Leonardo

Configuration d'Arduino



Choisir le port série correspondant
Ca peut ne pas être COM1 !

Arduino IDE



Arduino IDE 1.0.5

Langages

- C/C++
- Assembler

Exemples

- Lancer arduino
- fichier -> Exemples -> PRismino
- Permet de comprendre l'usage des fonctions

Programmation

Arduino “style”

```
#include <prismino.h>

void setup()
{
  // initialization
}

void loop()
{
  // toujours
}
```

C “normal”

```
#include <prismino.h>

int main(void)
{
  // initialization

  while(1)
  {
    // toujours
  }

}
```

Programmation - Example

Librairie

```
#include <prismino.h>

void setup()
{
  // set pin output mode
  pinMode(LED, OUTPUT);
}

void loop()
{
  // turn LED on
  digitalWrite(LED, HIGH);
  // wait 500 milliseconds
  delay(500);
  // turn LED off
  digitalWrite(LED, LOW);
  delay(500);
}
```

appelé une fois (initialisation)

Nous autorise à controler l' état du pin

allume la LED

appelé en boucle

attendre 500ms

éteint la LED

Programmation - ATTENTION

pinMode(pin N°,OUTPUT/INPUT);

selectionne l'état du pin (sortie/output/1 ou entrée/input/0)

- Ne pas le faire proprement peut endommager le PRismino!
- Au mieux, le programme ne fonctionne pas correctement

- **Dans le Setup, toujours configurer les PINs utilisés!**

N.B. À la base, les PINs sont en entrée (0)

Illustration!

- Gamme de fréquence du buzzer
- Lier buzzer et potentiometre

Des questions?

Hésitez pas à venir nous voir!
Il y a encore des kits en vente!

Infos!

- samedi 1 novembre: **Journée de montage**
  - En haut du BM, début à 8h
  - Jusqu'à 17h
  - Comités disponibles
  - Repas 5.-
  - Possibilité d'acheter un kit/s'inscrire

- lundi 20 octobre: **Démon Alimentation, PWM, moteurs**