Bouton et capteur IR

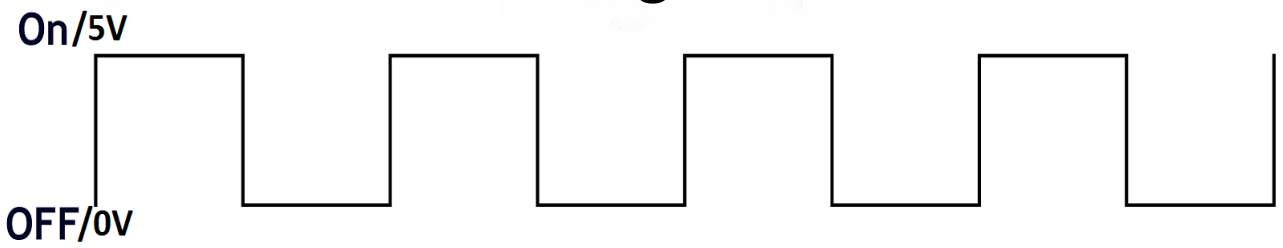
Comment “voir et sentir” le monde qui nous entoure

Déroulement du démon

-
- digital/analogue
- bouton
- capteur IR
- utilisation des pins
- principes de programmation
- applications

Signal digital/analogue

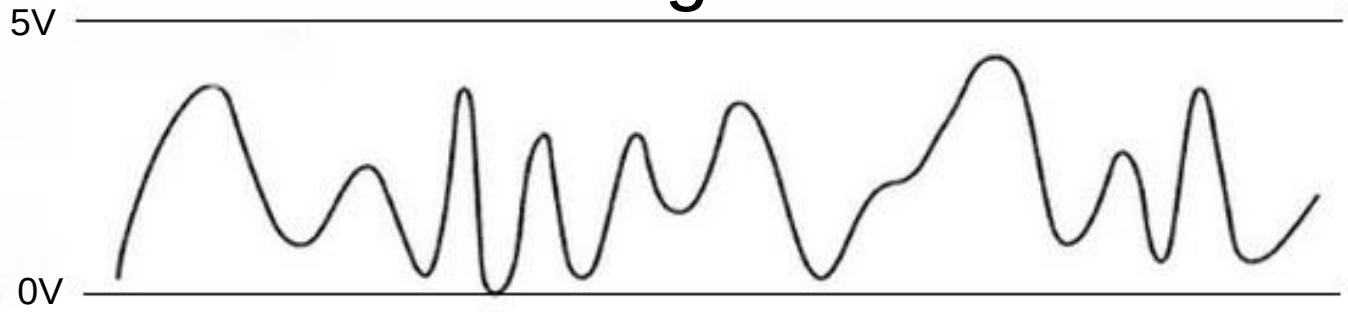
Digital



Faux = 0V

Vrai = 5V

Analogue

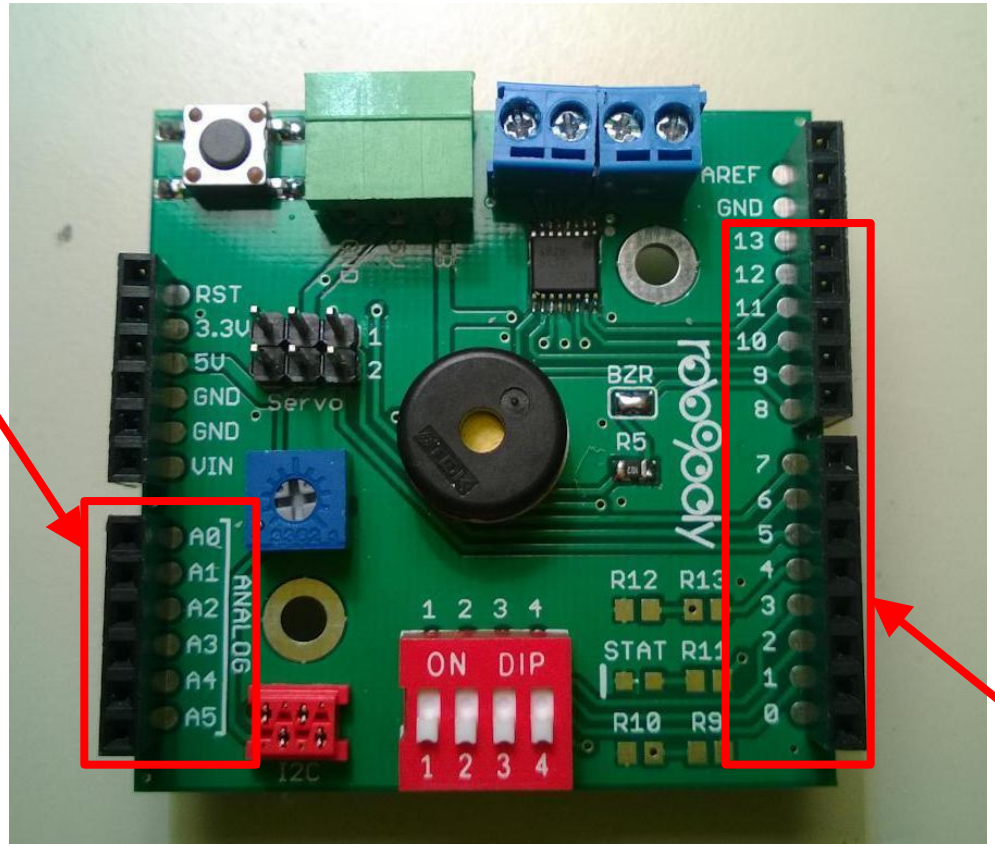


n'importe quelle valeur

Pin digital/analogique

Pins “analogiques” (A1-A5)

peuvent aussi être utilisé pour lire en digital, mais du coup on prends de la place sur les pin analog. pour rien



Pins “digitals” (0-13)

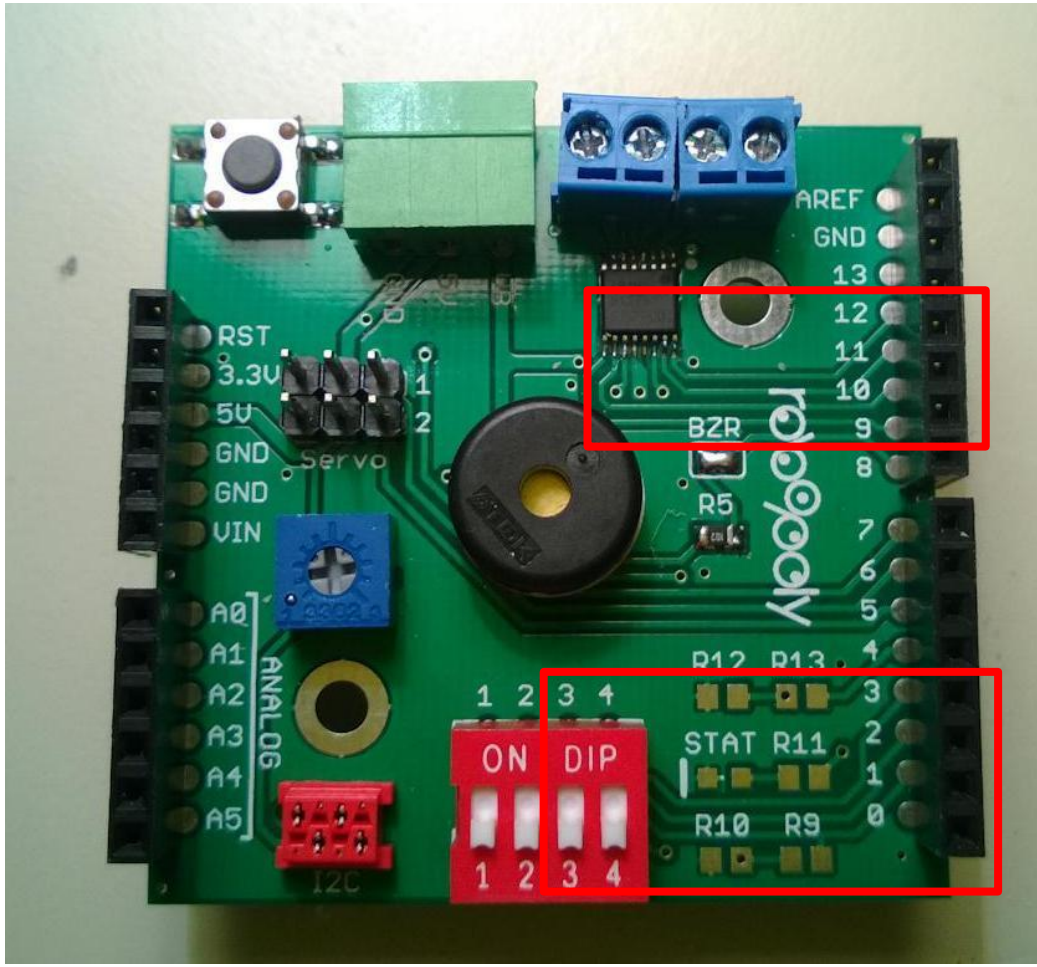
Avant de faire des bêtises

vérifier que le pin soit libre !

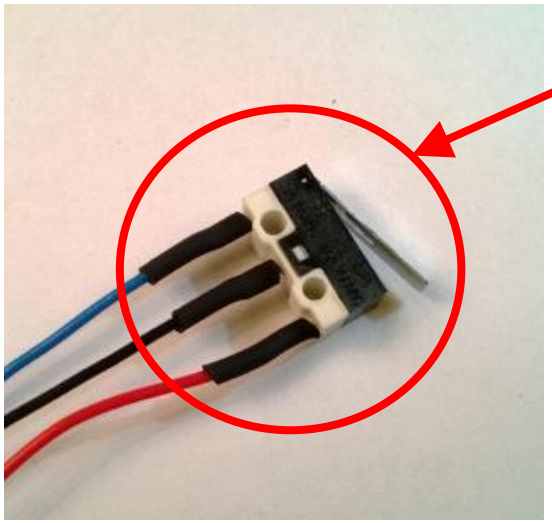
Comme le shield comporte plusieurs accessoires (servo, pont-H,...) certains pins sont justement utilisés pour leur fonctionnement et ne peuvent pas être utilisés pour brancher n'importe quoi ! On peut p.ex. voir que les pistes du pont-H utilise les pin de 8 à 12, donc pas touche !

Voici la répartion des pin utilisables :

- A0 : potentiomètre
- A1, A2, A3 : capteur IR
- A4, A5 : bouton poussoir
- 1 - 2 - 3 - 4 : DIP switch
- 13 : rail d'alim des capteurs

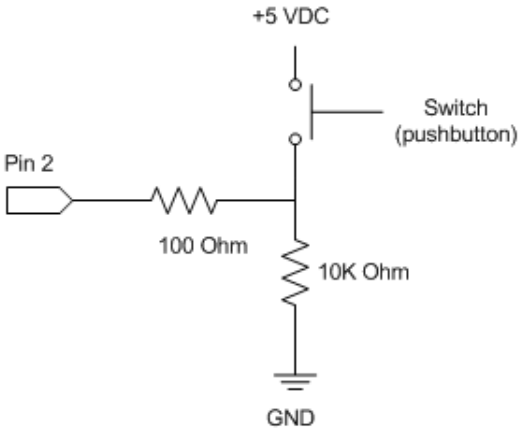


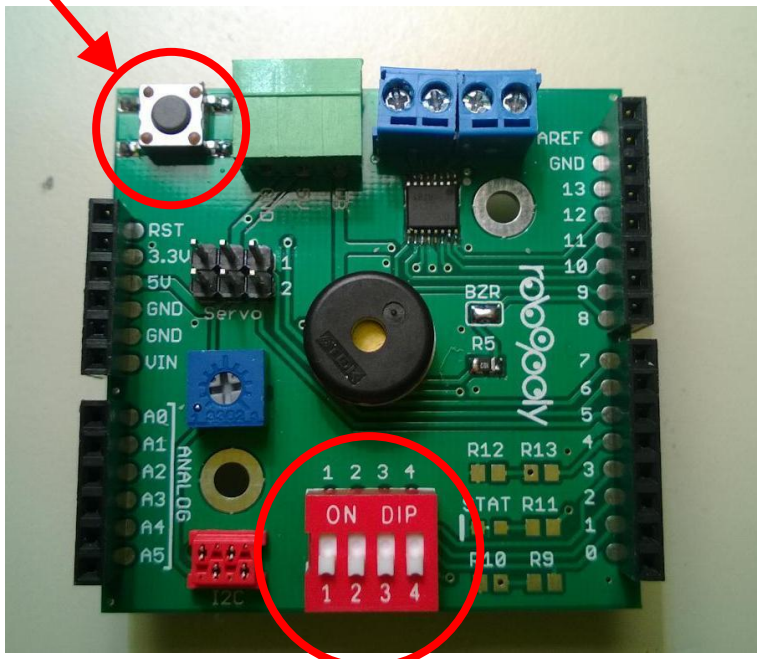
Bouttons/Switches



Bouton

Pushbutton Schematic

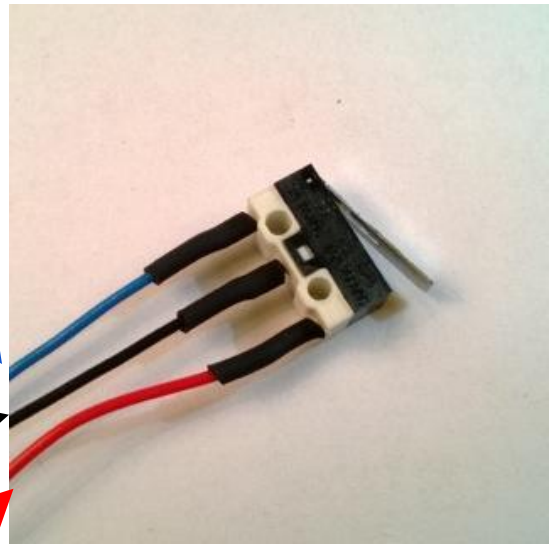




DIP-Switch

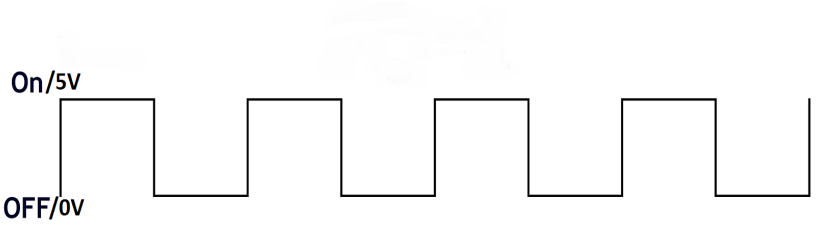
Fonctionnement du button

Signal
ouvert = 5V
fermé = 0V
(digital)



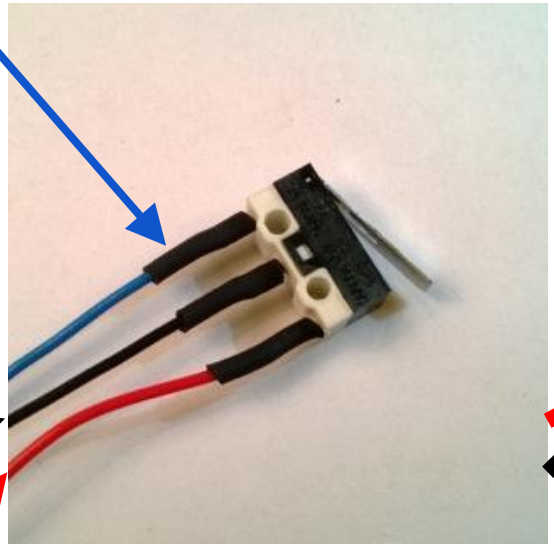
GND

VCC



Fonctionnement du button

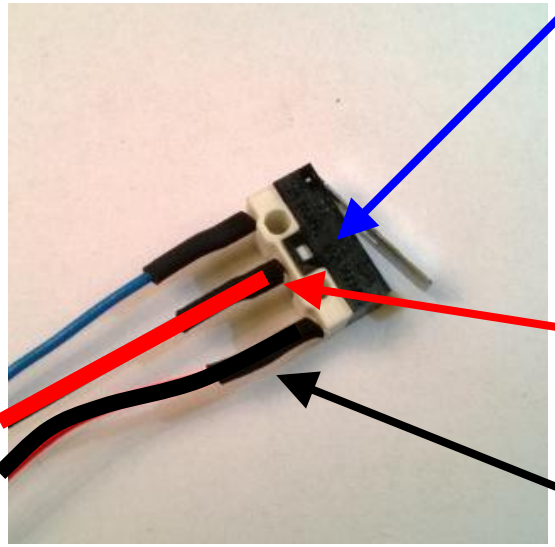
Signal
ouvert = 5V
fermé = 0V
(digital)



GND

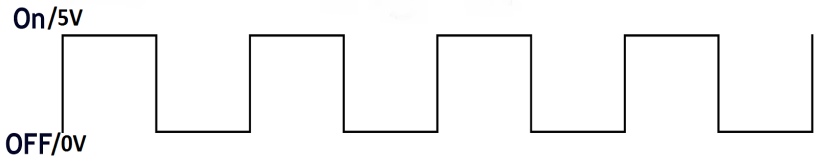
VCC

Signal
ouvert = 0V
fermé = 5V
(digital)



VCC

GND



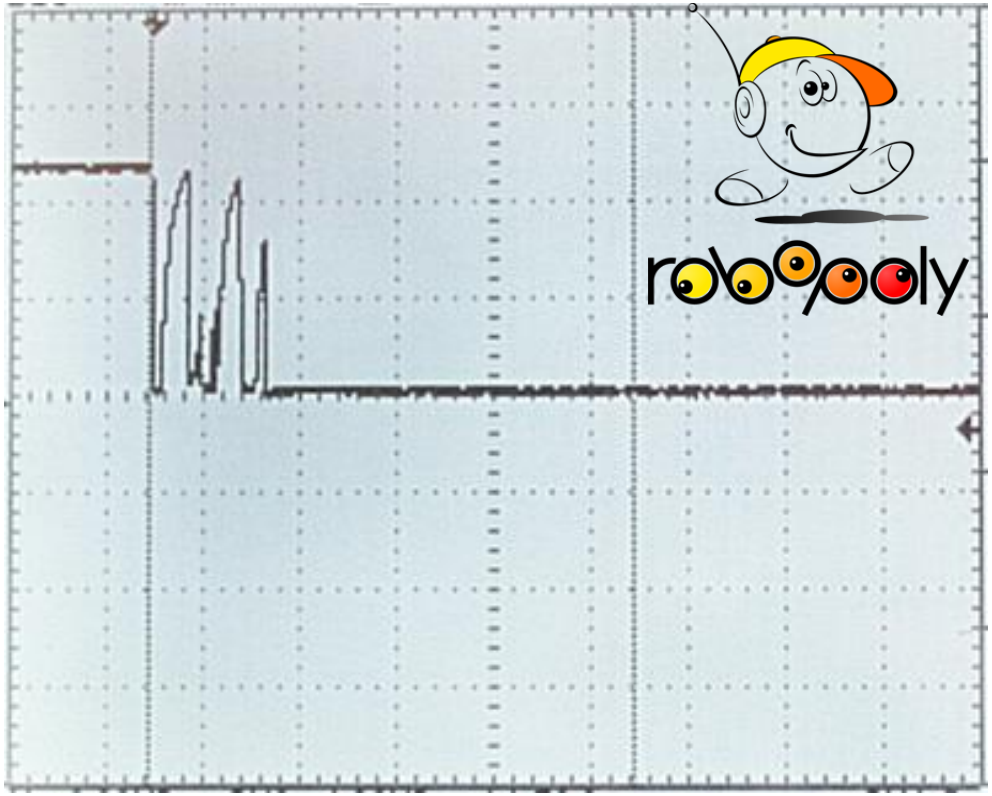
Debounce

Quand on appuie sur un bouton, le passage de 0 à 1 ne se fait pas instantanément, il y a une oscillation avant de changer d'état.

```
if(capteur == 1)
        // allumer LED
else if(capteur == 0)
        // éteindre
```

LED

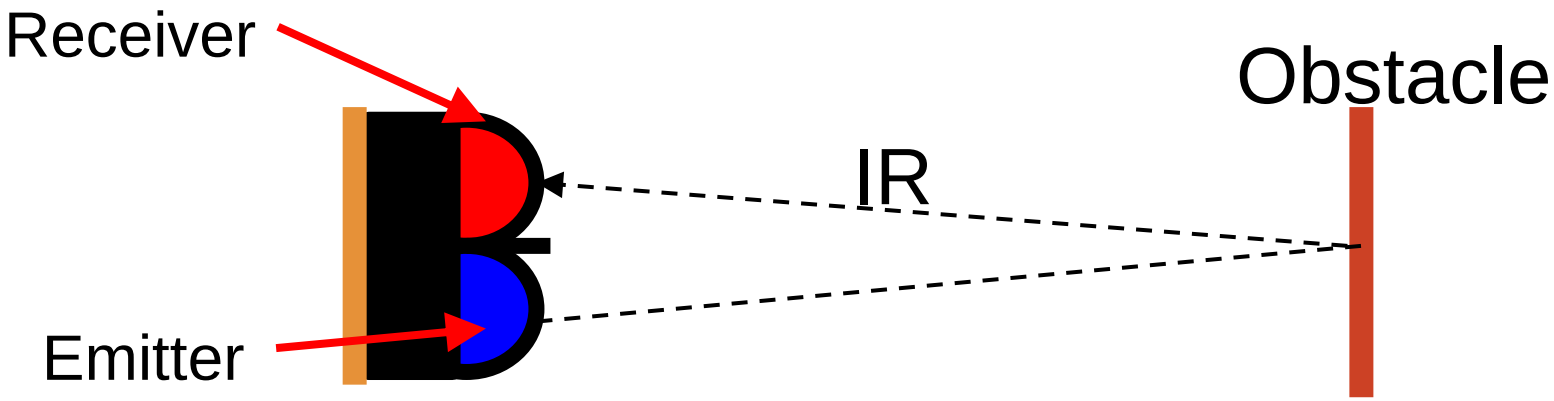
En effectuant le code ci_dessus, la LED risque de clignoter avant de rester allumée quand le capteur passe de 0 à 1. (peut-être que ça ne sera pas visible à l'oeil nu mais si c'est autre chose qu'une LED, ça risque de poser problème)



Résolution :

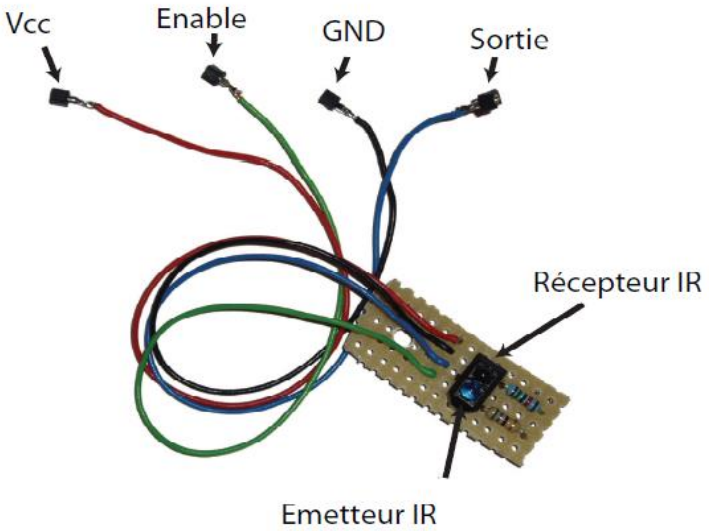
- ajouter une capacité
- faire deux mesures sur un intervalle de temps pour voir si la valeur à changé (il y a une fonction arduino pour ça)
- mettre un temps de pause avant chaque mesure à la fin de la boucle avec delay(100);

Capteur IR (infrared)

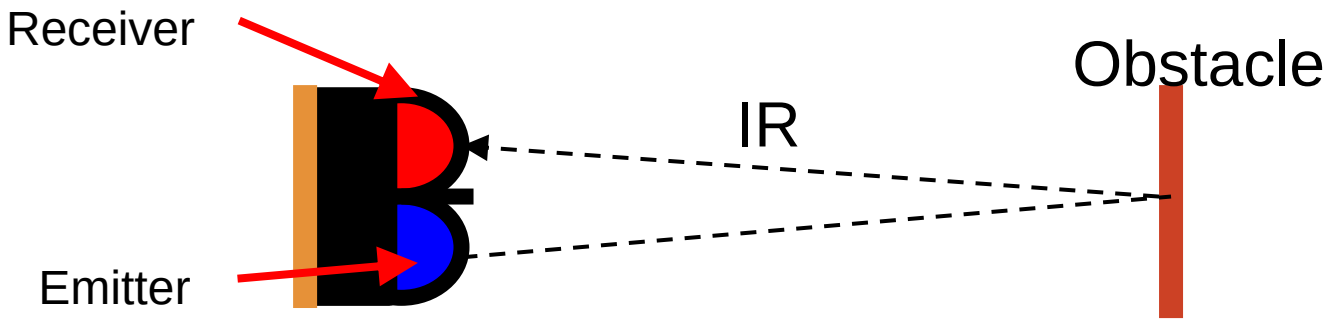


deux modes de fonctionnement:

- digital (0 ou 1)
- analogue (0 - 1023)

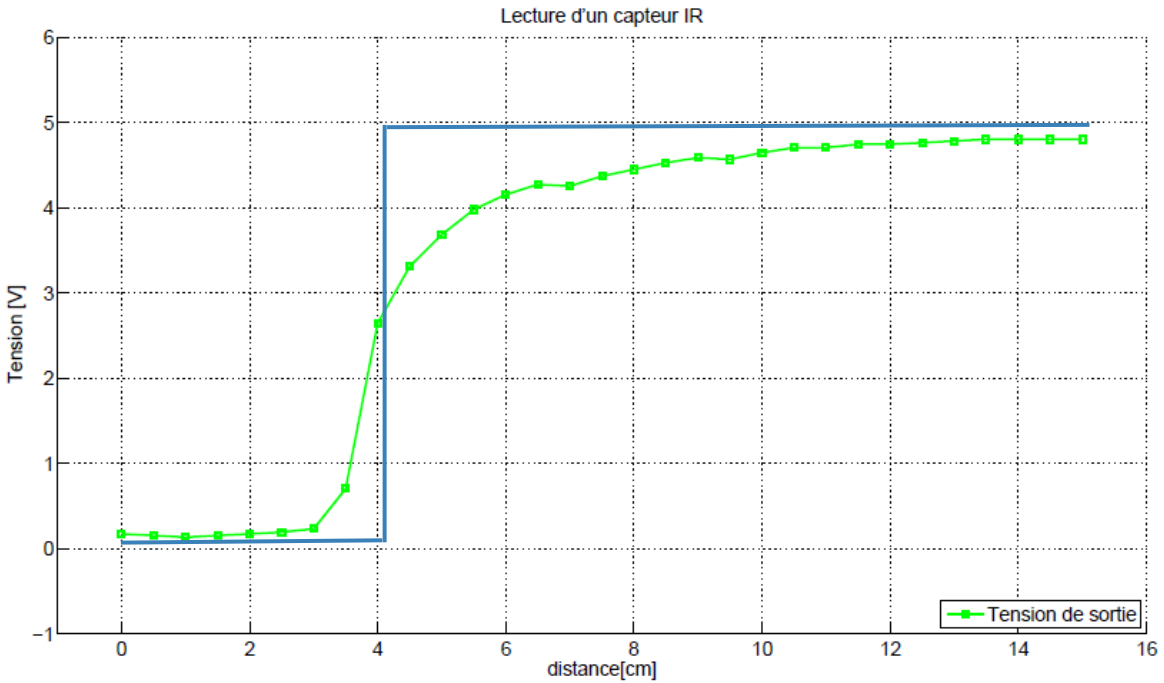


Capteur IR - fonctionnement

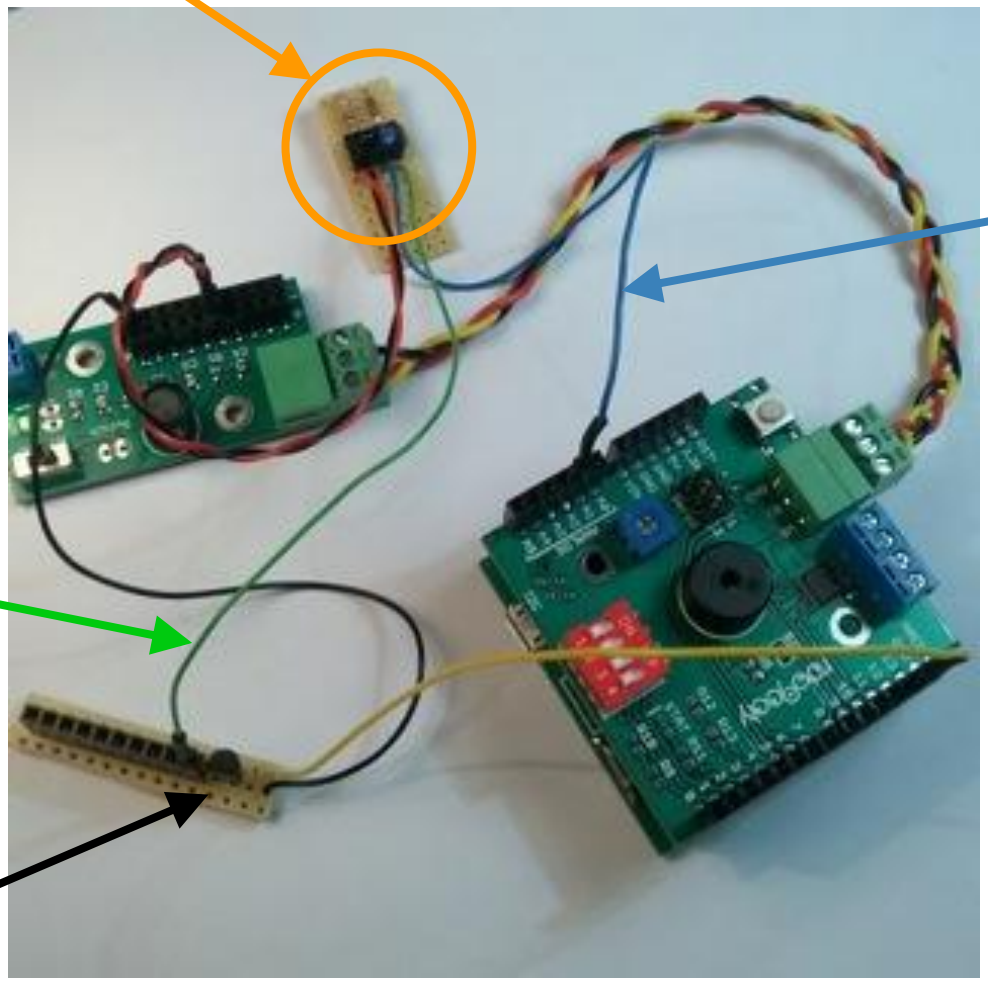


deux modes de fonctionnement:

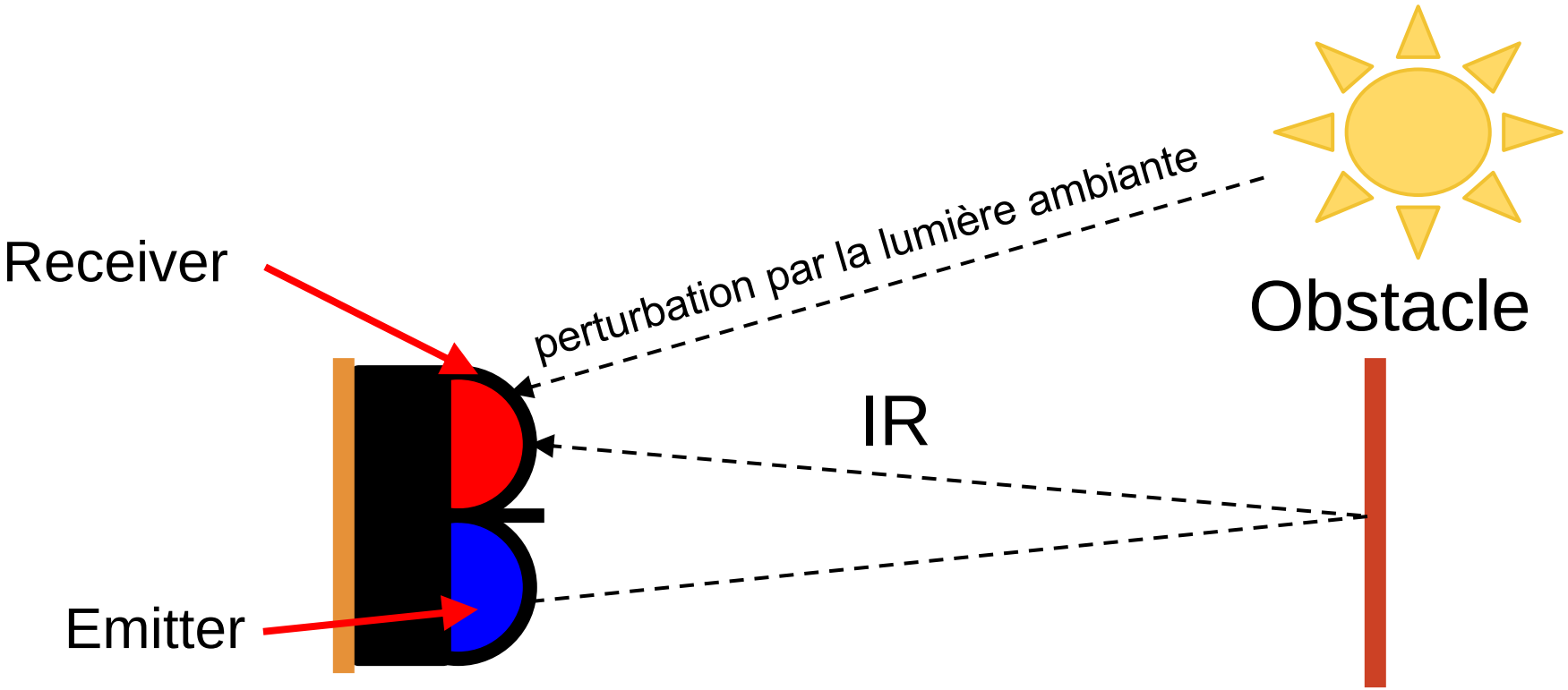
- digital (0 ou 1)
- analogue (0 - 1023)



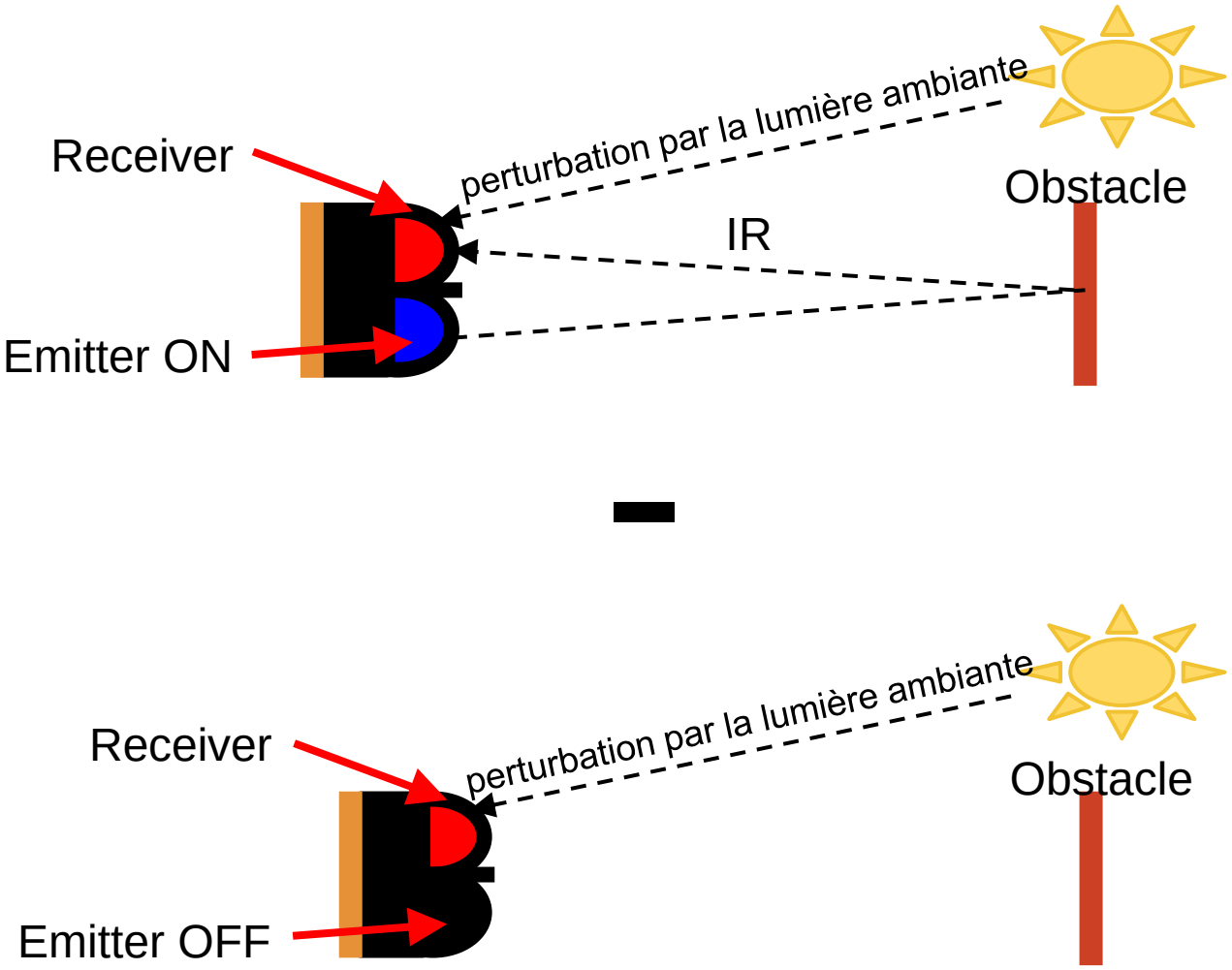
Capteur IR - branchement



Capteur IR - perturbations

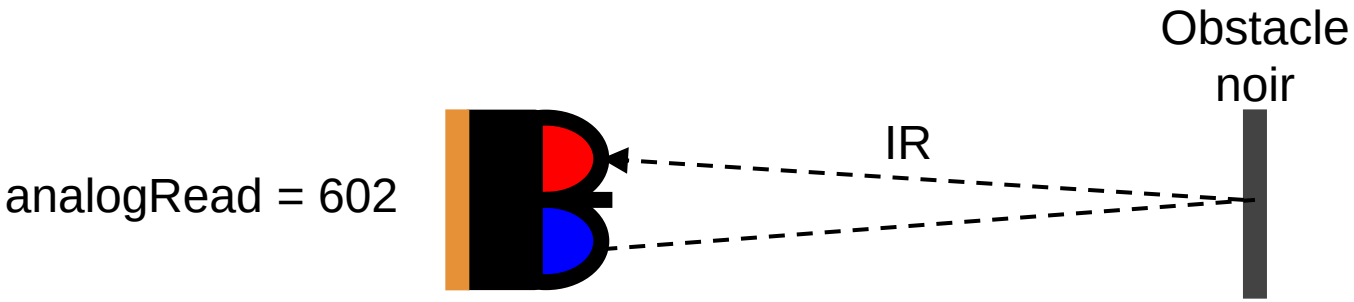


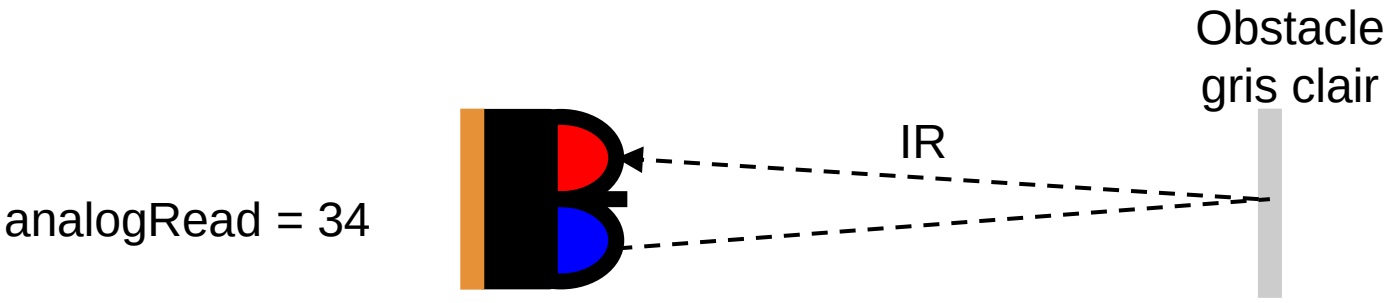
Capteur IR - perturbations

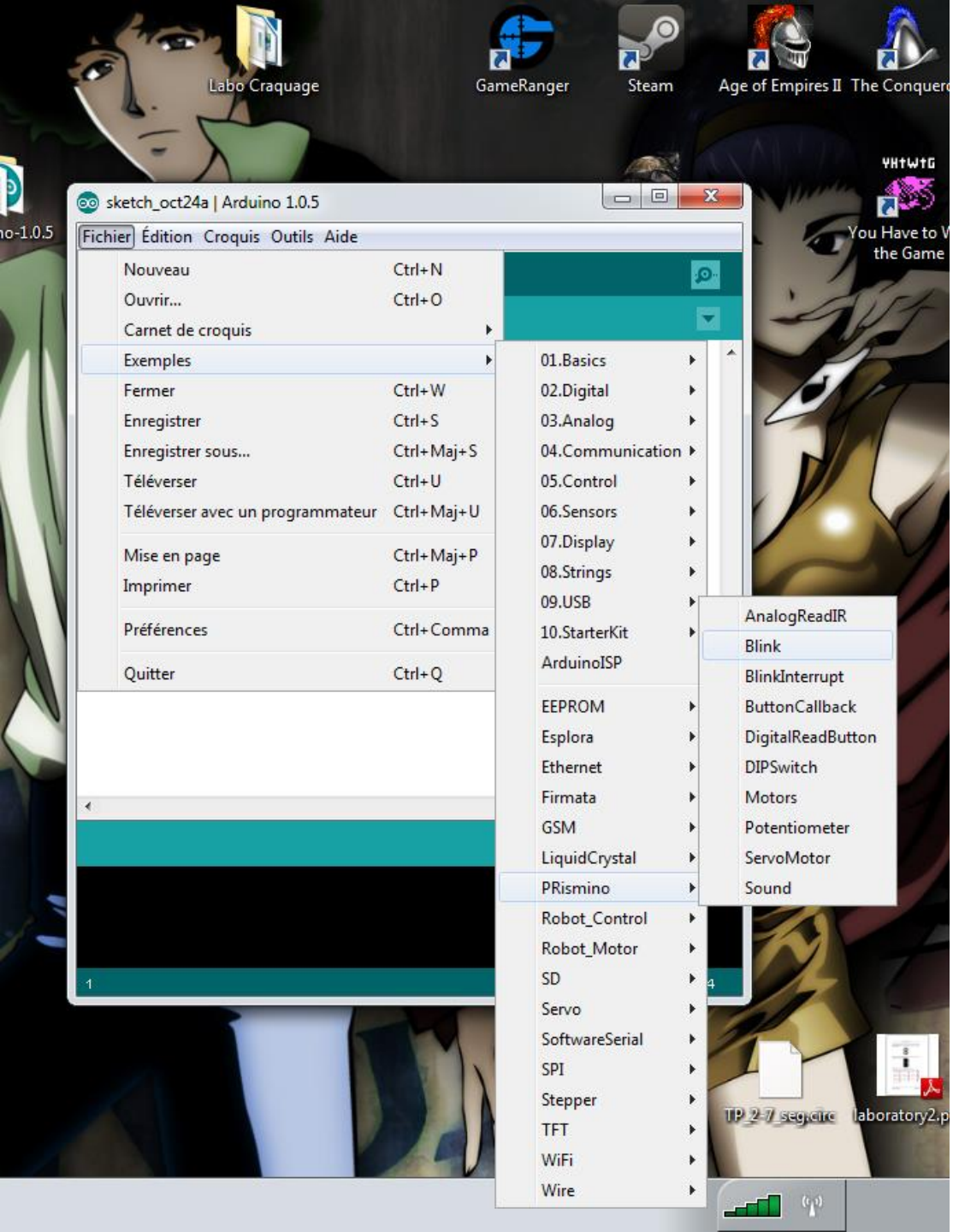


Capteur IR - surfaces

analogRead [0,1023]



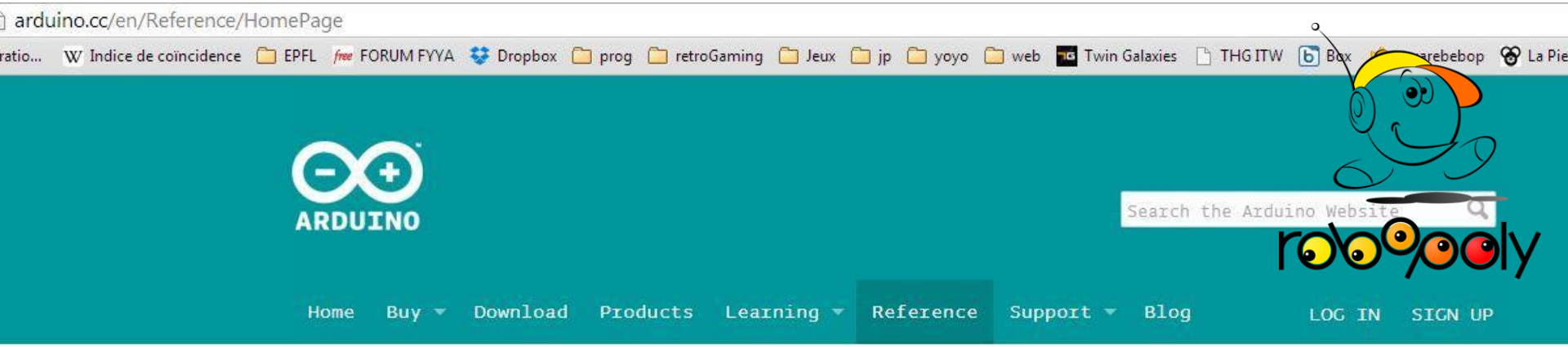




Regarder les exemples PRismino, c'est un bon moyen de se souvenir comment utiliser une fonction.

- Blink
- AnalogReadIR
- DigitalReadButton

http://arduino.cc/en/Reference/HomePage



Reference Language | Libraries | Comparison | Changes

Language Reference

Arduino programs can be divided in three main parts: *structure*, *values* (variables and constants), and *functions*.

Structure

- setup()
- loop()

Control Structures

- if
- if...else

Variables

Constants

- HIGH | LOW
- INPUT | OUTPUT| INPUT_PULLUP
- true | false
- integer constants

Functions

Digital I/O

- pinMode()
- digitalWrite()
- digitalRead()

Analog I/O

mode INPUT / OUTPUT

INPUT : lecture

OUTPUT : écriture

```
pinMode(pin, mode);
```

numéro du pin

INPUT ou OUTPUT

- INPUT : le pin est utilisé pour lire / récuperer des données (capteur, bouton,...)

- OUTPUT : le pin est utilisé pour communiquer / faire fonctionner des accessoires (LED, moteur, servo,...)

Pins configured as outputs can also be damaged or destroyed if short circuited to either ground or 5 volt power rails.

mode INPUT / OUTPUT

```
#include <prismino.h>

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

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

On met la LED en mode sortie (OUTPUT) pour lui donner une valeur (HIGH ou LOW).
On ne va quand même pas essayer de récupérer la valeur de la LED...

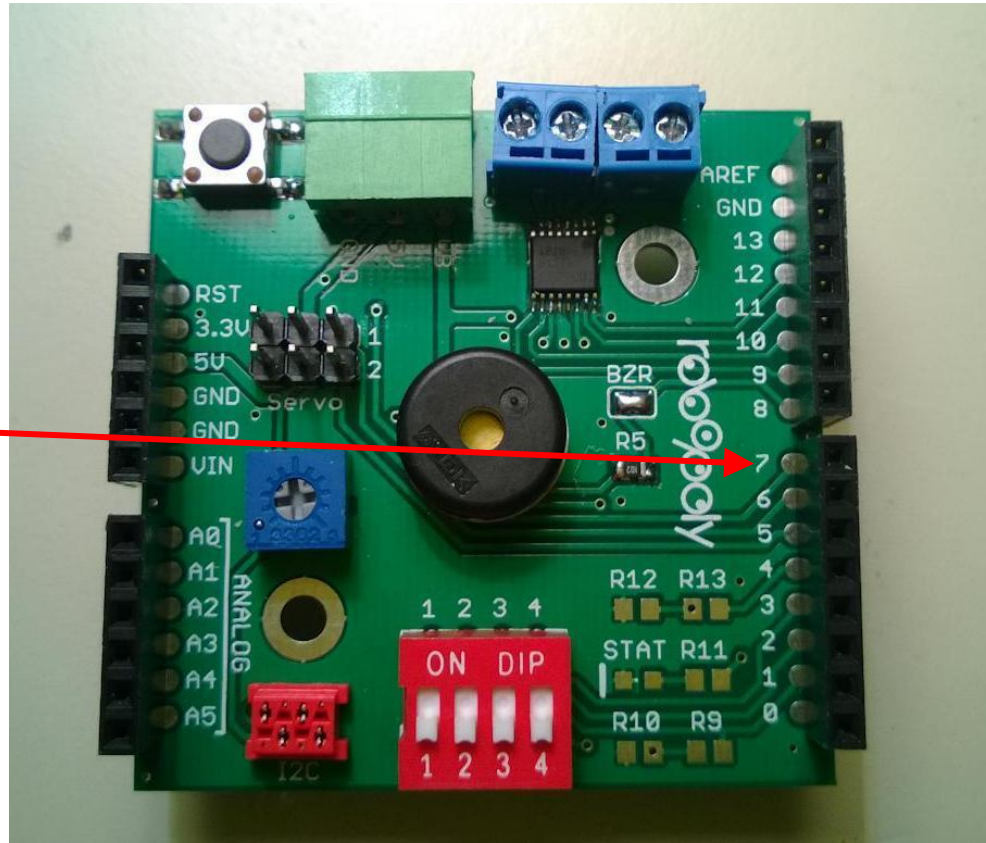
digitalWrite

écrire en digital

HIGH / LOW

digitalWrite(pin, value);

numéro du pin
p. ex. 7



digitalRead

lire en digital

digitalRead(pin);

numéro du pin
p. ex. 3

```
#include <prismino.h>

// the button value can be 0 or 1
unsigned char button_value;

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

void loop()
{
  // connect lever button to pin A4
  button_value = digitalRead(A4);

  if(button_value)
  {
    digitalWrite(LED, HIGH);
  }
  else
  {
    digitalWrite(LED, LOW);
  }
  delay(100);
}
```

analogRead

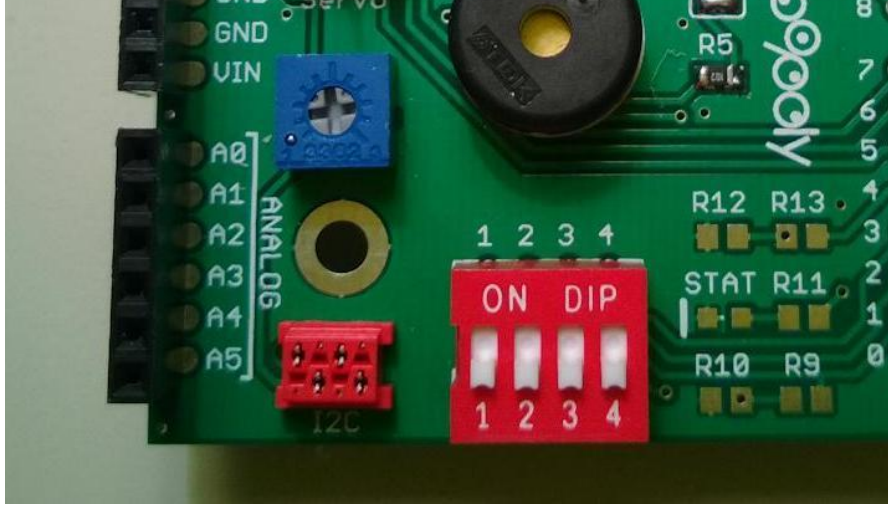
analogRead(pin);

lire en analogique, principalement utilisé pour lire une valeur analogique des capteurs IR
valeur retournée entre **0 et 1023**

```
#include <prismino.h>

// the analog value is between 0 and 1023 so it needs a 16-bit variable
unsigned int ir_value;

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

void loop()
{
  // connect sensor to pin A1, A0 is used by the potentiometer on the shield
  ir_value = analogRead(A1);

  if(ir_value < 512)
  {
    // a low value means something (an obstacle) reflects emitted IR to the receiever
    digitalWrite(LED, HIGH);
  }
  else
  {
    // no obstable is detected, turn LED off
    digitalWrite(LED, LOW);
  }
  delay(100);
}
```

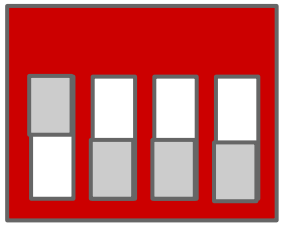
DIP switch

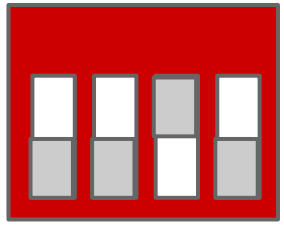
2^4=16

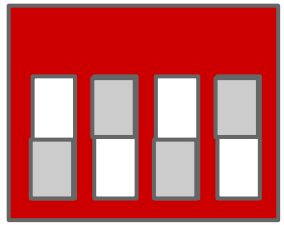


Le DIP switch est un moyen d'interagir avec le code en lui communiquant des informations.

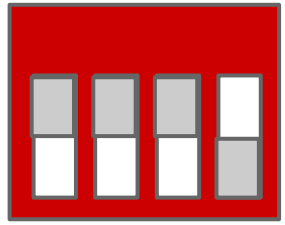
On a 4 switch avec 2 positions, donc on peut au maximum communiquer 2^4 positions soit 16 !
On peut donc soit les utiliser "un par un", ce qui fait 4 boutons, soit comme un nombre binaire !







0101 -> 5



1110 -> 14

"un par un", on utilise seulement le bouton voulu

binaire : on "écrit" un nombre binaire qui sera entre 0 et 15

switch case

tester plusieurs cas

On peut tester la valeur retournée par un capteur IR en analogique ou la valeur du DIP-switch p.ex.

```
switch (variable)
{
  case 42:
    //je fais ça
    break;
  case 3:
    //ou plutôt ceci
    break;
  case -273:
    //en fait non, c'est mieux ça
    break;
  case obi-wan kenobi:
    //ouais non, ce cas est mieux
    break;
}
```

Infos!

- Journée montage samedi 09/11/2013
  - En haut du BM, début à 9h
  - Jusqu'à 17h
  - Comités disponibles
  - Repas 5.-
  - Possibilité d'acheter un kit/s'inscrire

- Prochain démon : Alessandro Crespi du laboratoire BioRob -> salamandre