



robot

## Boutons et capteurs IR

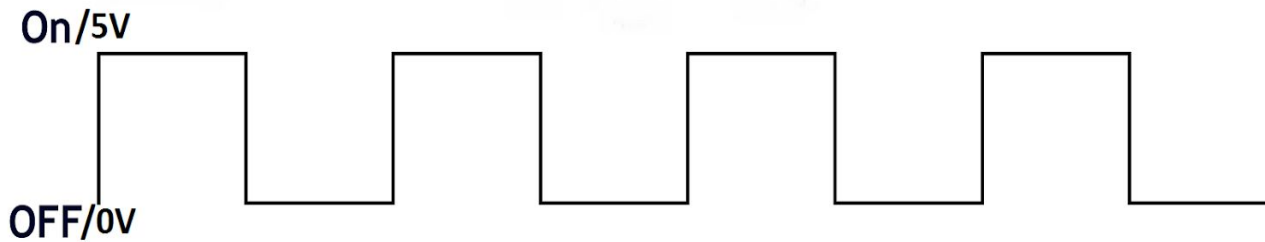
Comment voir et sentir le monde qui nous entoure

# Qu'est ce que l'on apprend ?

- Monde digital / analogique (Rappel)
- Bouton
- Capteur IR

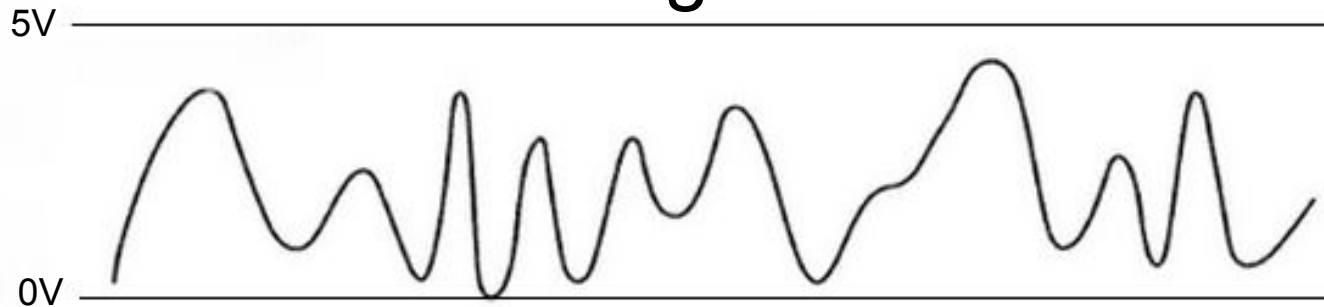
# Signal digital/analogique

## Digital



Faux = 0V  
Vrai = 5V

## Analog

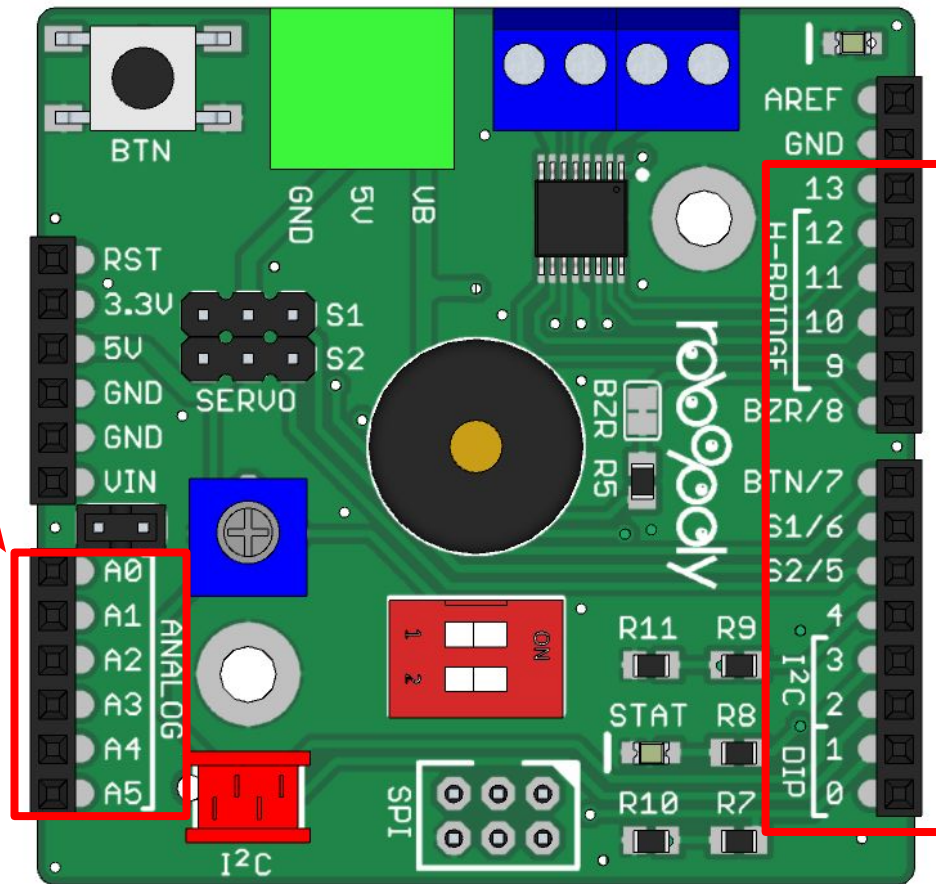


n'importe  
quelle  
valeur

# Pin digital/analogique

**Pins  
“analogiques”  
(A1-A5)**

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



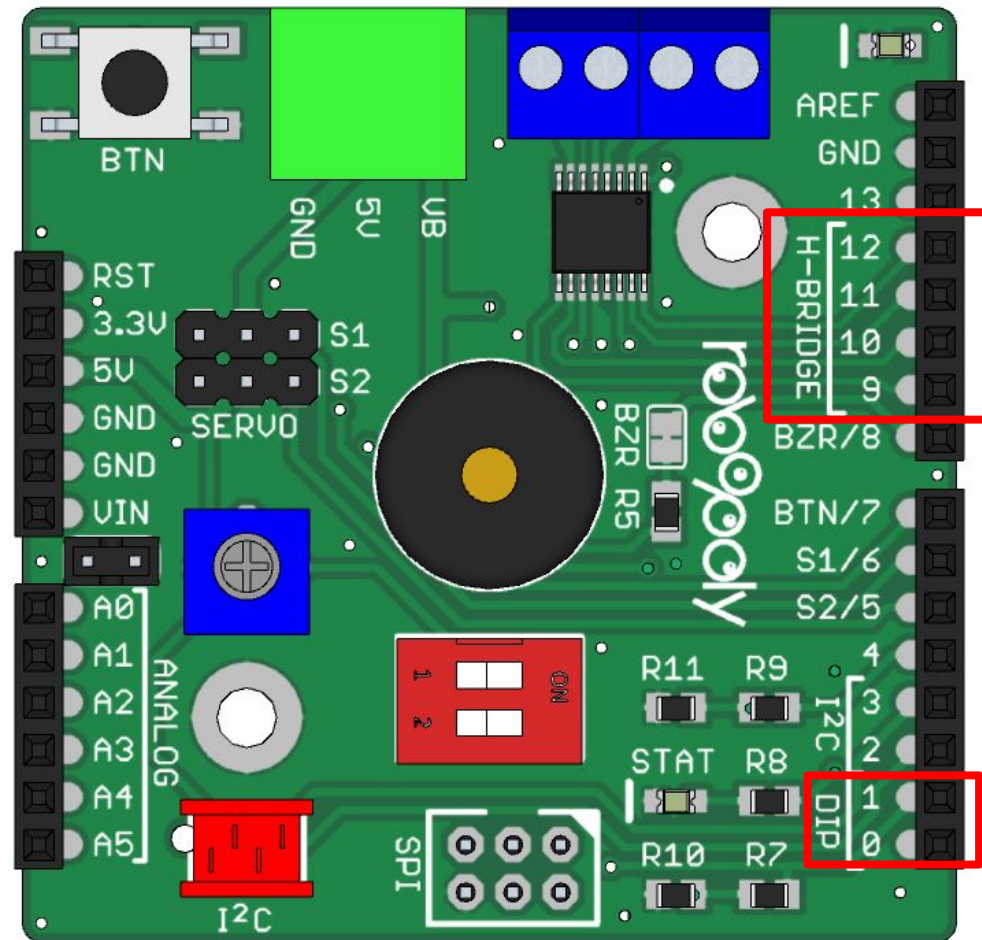
**Pins  
“digitals”  
(0-13)**

# Avant de faire des bêtises

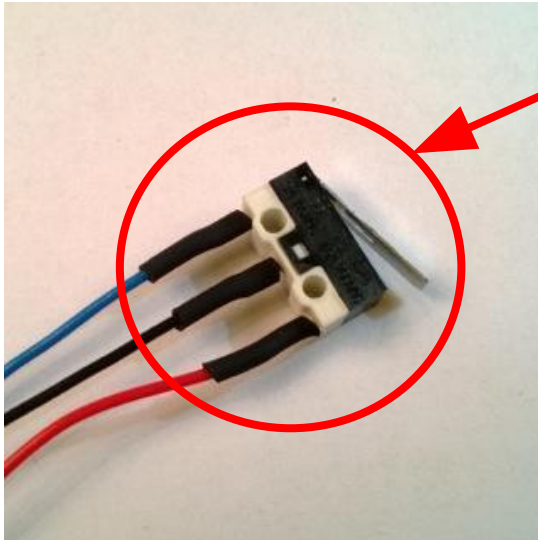
vérifier que le pin soit libre !

Comme le shield comporte plusieurs accessoires (pont-H,...) certains pins ne sont pas disponibles

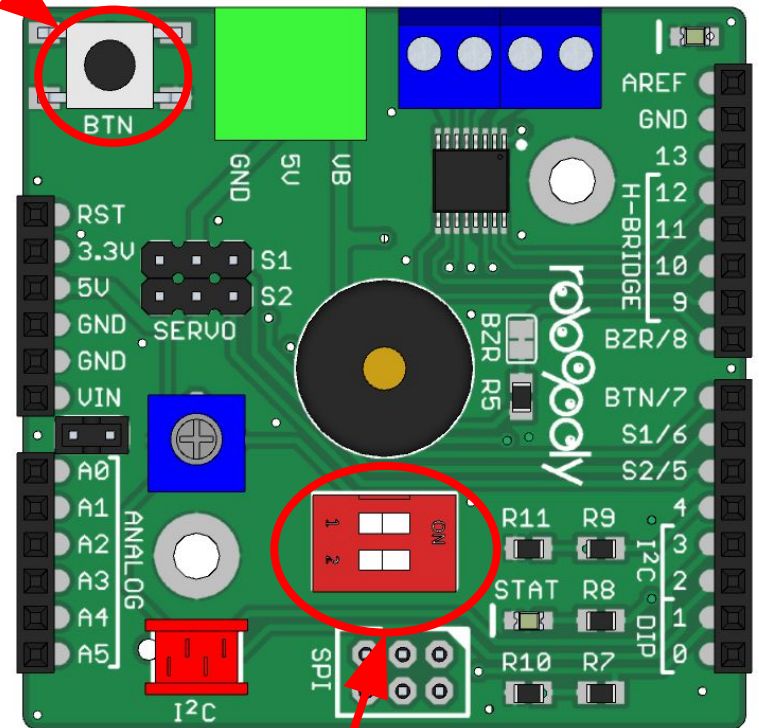
Voir le [site](#) pour savoir lesquels utiliser!



# Buttons/Switches



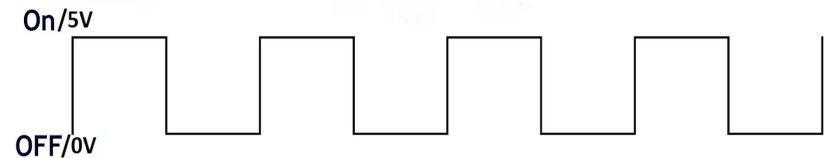
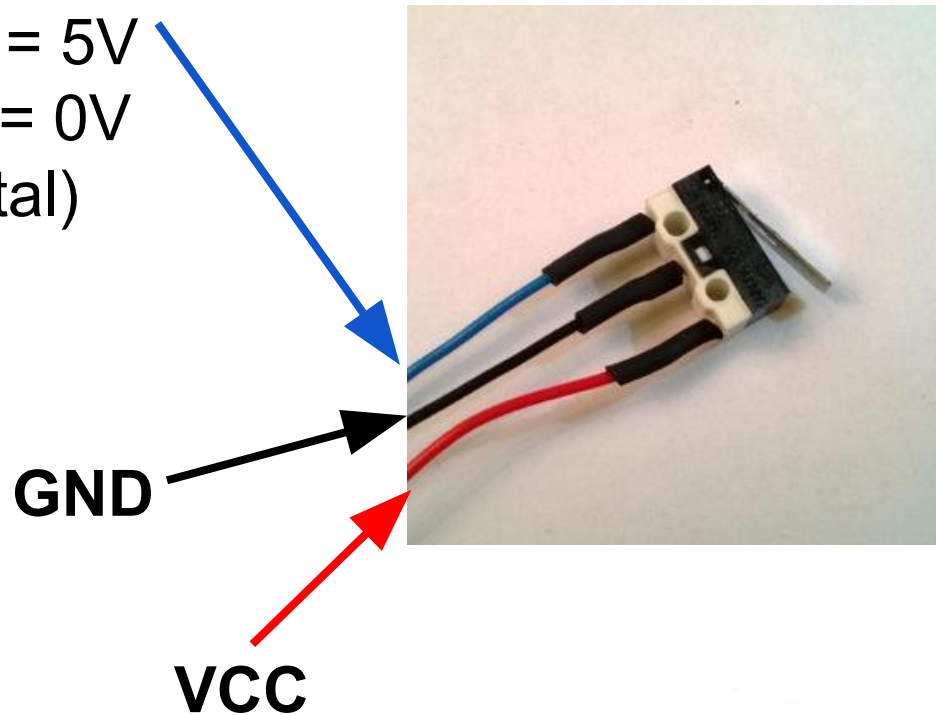
Bouton



DIP-Switch

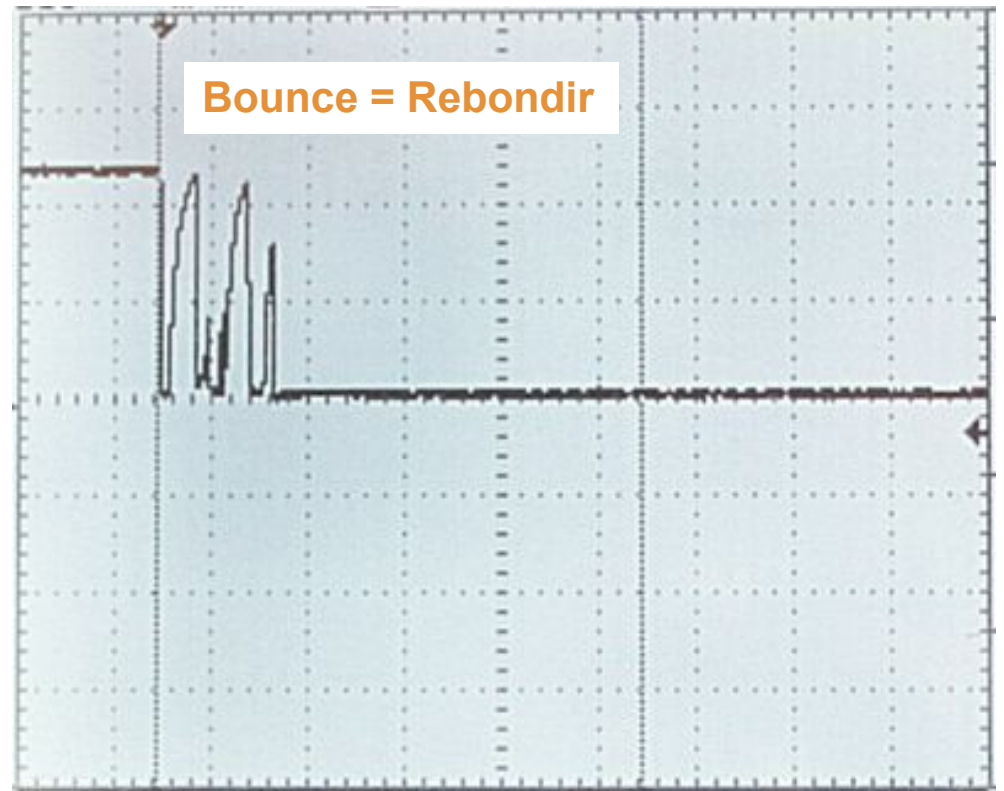
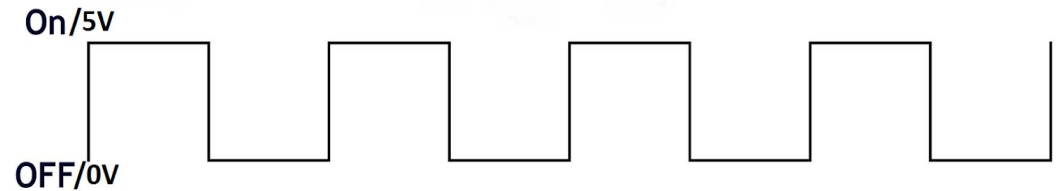
# Fonctionnement du bouton

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



# Debounce

Quand on appuie sur un bouton, la mécanique cause des oscillations dans la tension lue, vous risquez donc d'avoir un comportement incertain!

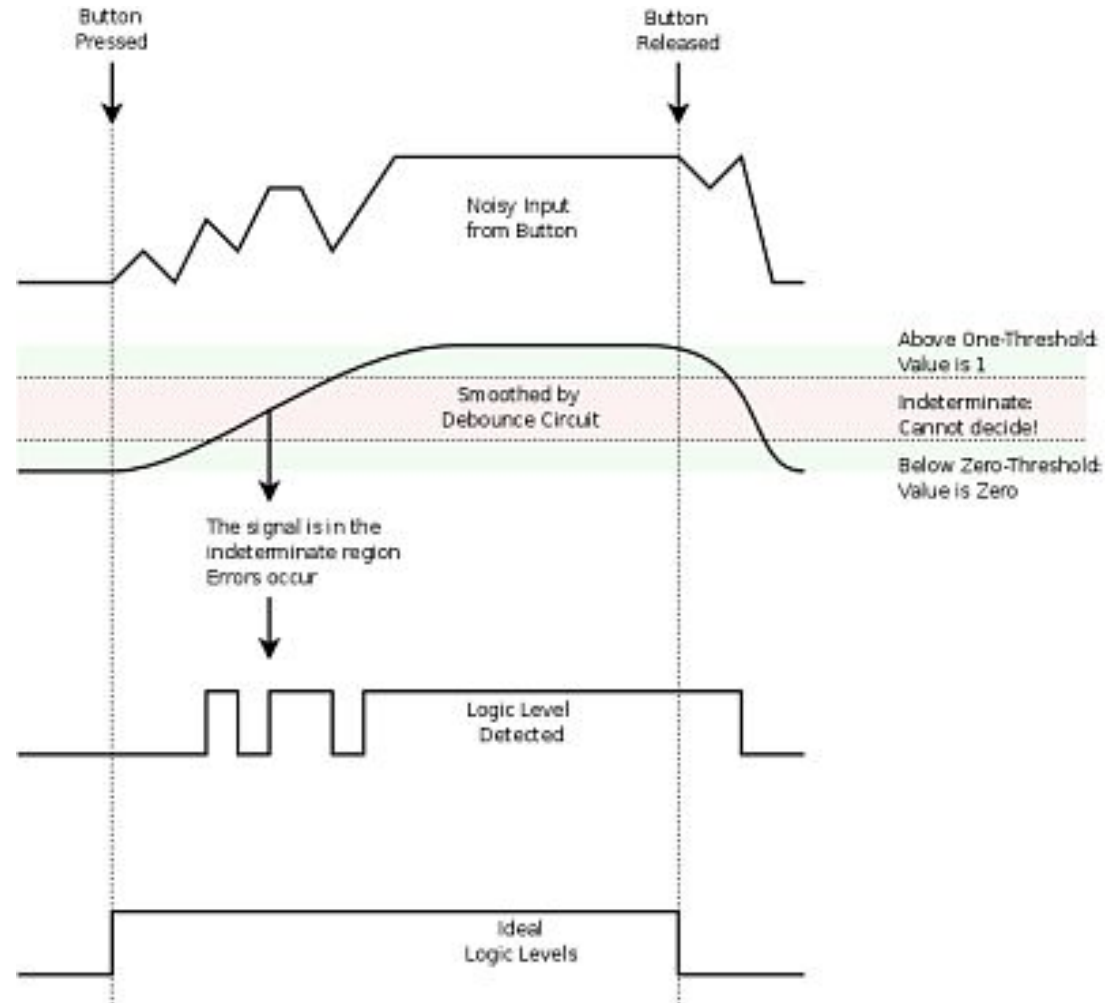
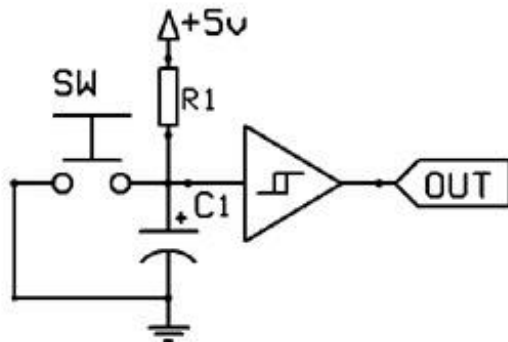




# Debounce

Solutions :

- 1) Ajouter une capacité (et une résistance)
- 2) Faire deux mesures sur un intervalle de temps pour voir si la valeur à changé.
- 3) Faire des circuits de debounce avec des bascules



# Programmation

- Input ou input pull-up
- DigitalRead()
- Attention au bounce

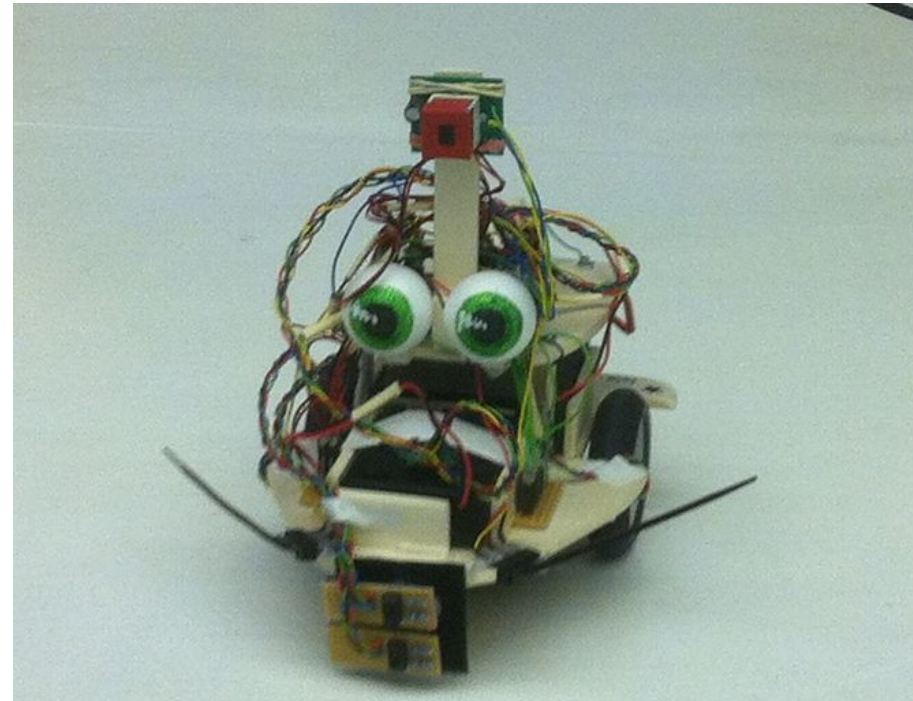
```

1  #include <prismo.h>
2
3  #define BUTTON_PIN 5
4
5  int button;
6
7
8  void setup() {
9      pinMode(BUTTON_PIN, INPUT);
10 }
11
12 void loop() {
13     button = digitalRead(BUTTON_PIN);
14     if(button == 1)
15     {
16         // do something here
17     }
18     else
19     {
20         // do something else
21     }
22 }

```

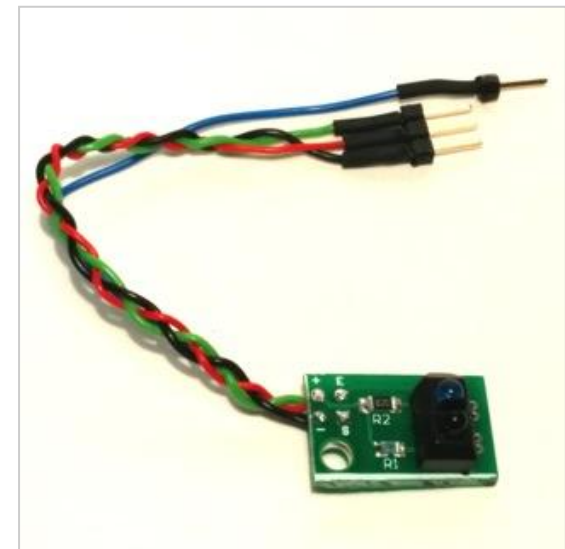
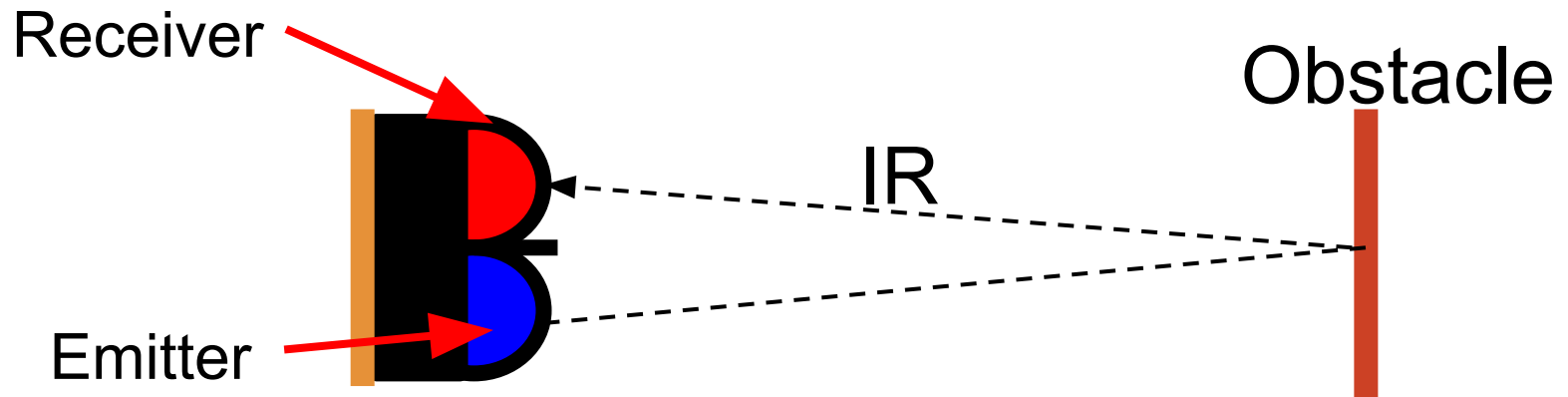
# Utilisation

- Détecter un obstacle
- Réaliser un “reset”
- Jeux
- ...



**Show time!**

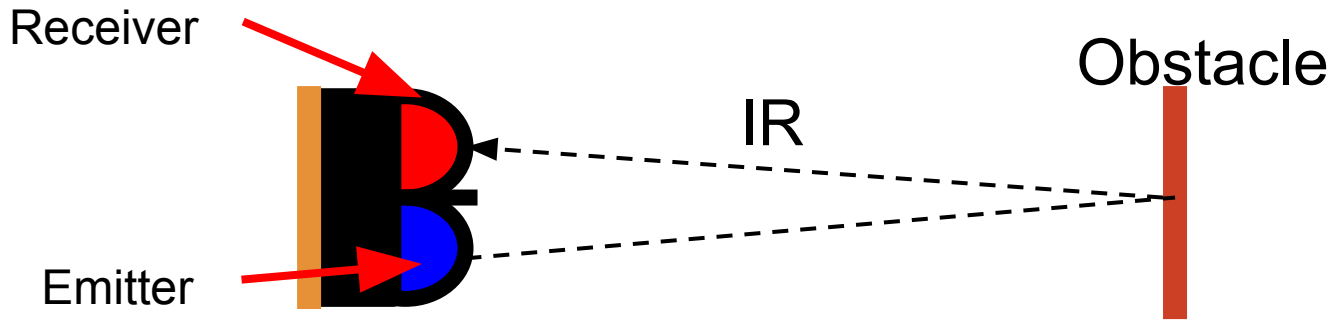
# Capteur IR (infrarouge)



deux modes de fonctionnement:

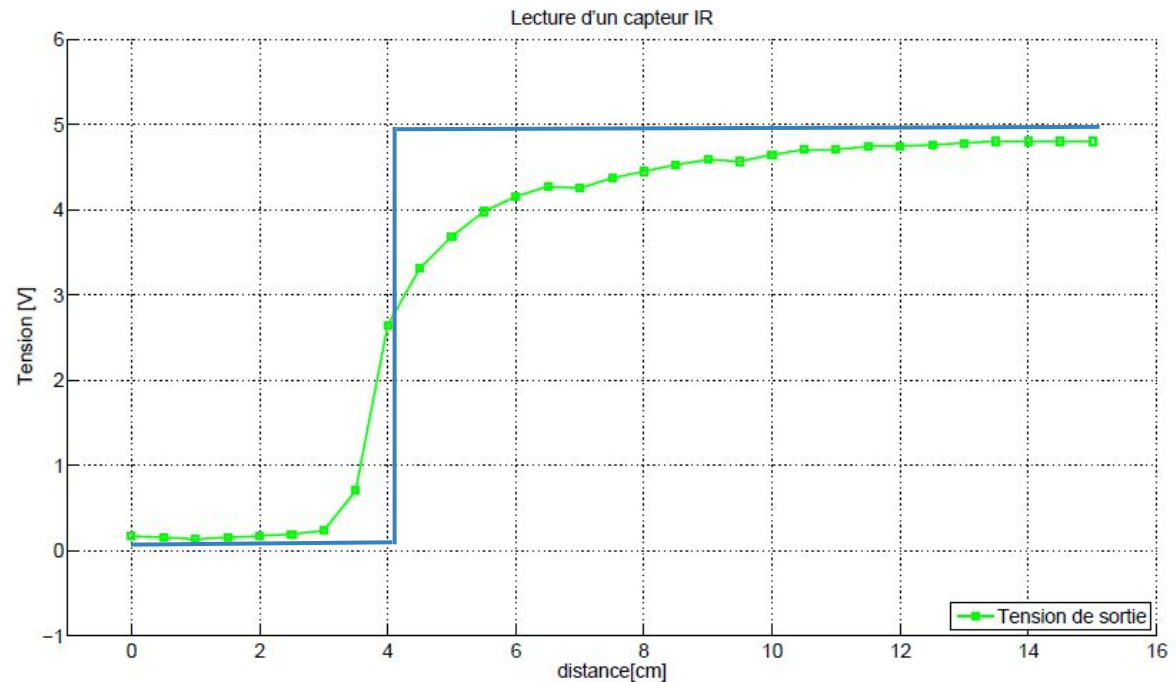
- digital (0 ou 1)
- analogique (0 - 1023)

# Capteur IR - fonctionnement

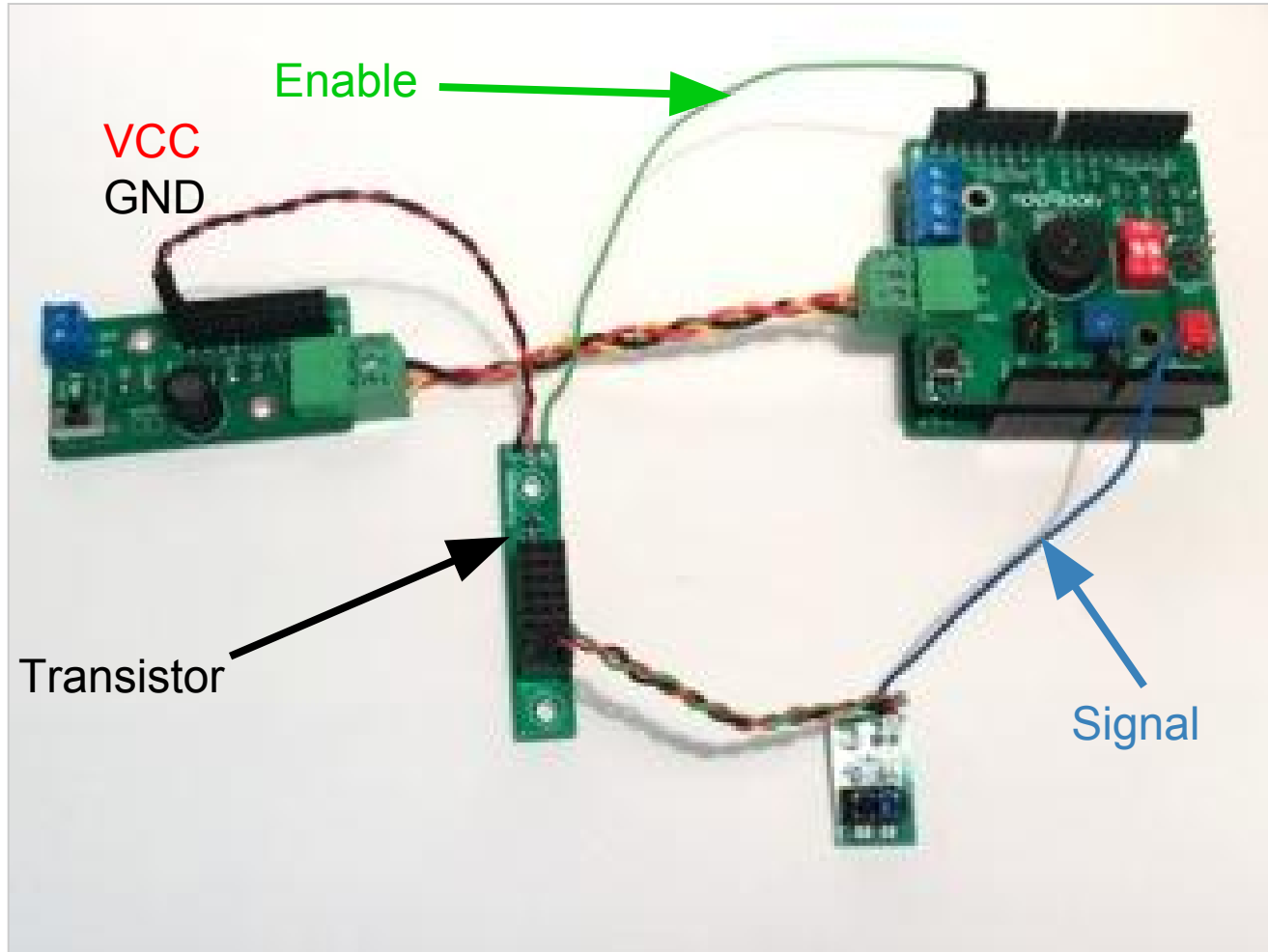


deux modes de fonctionnement:

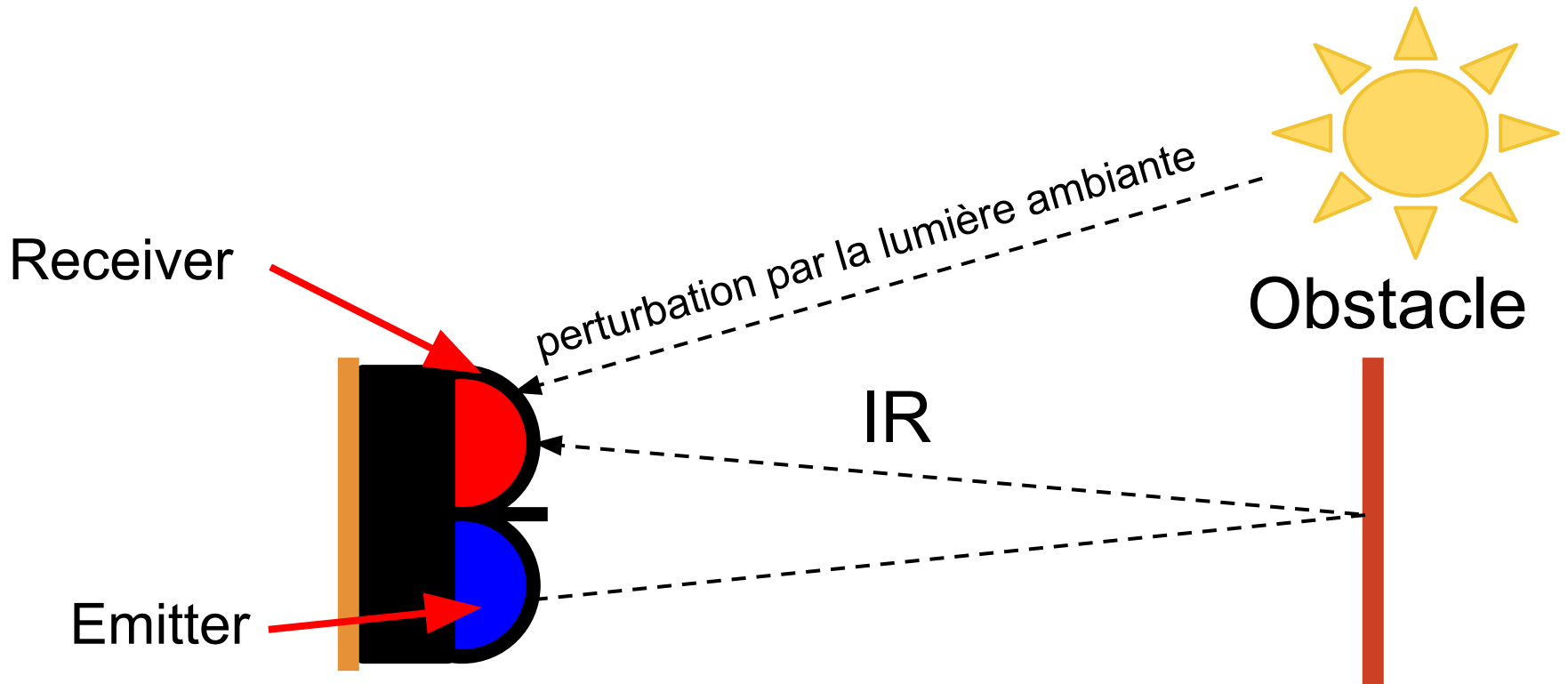
- digital (0 ou 1)
- analogique (0 - 1023)



# Capteur IR - branchement

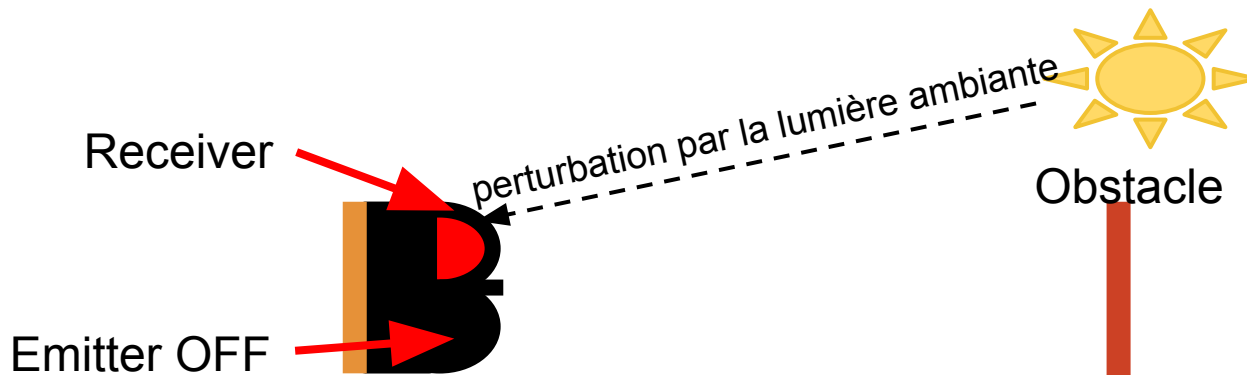
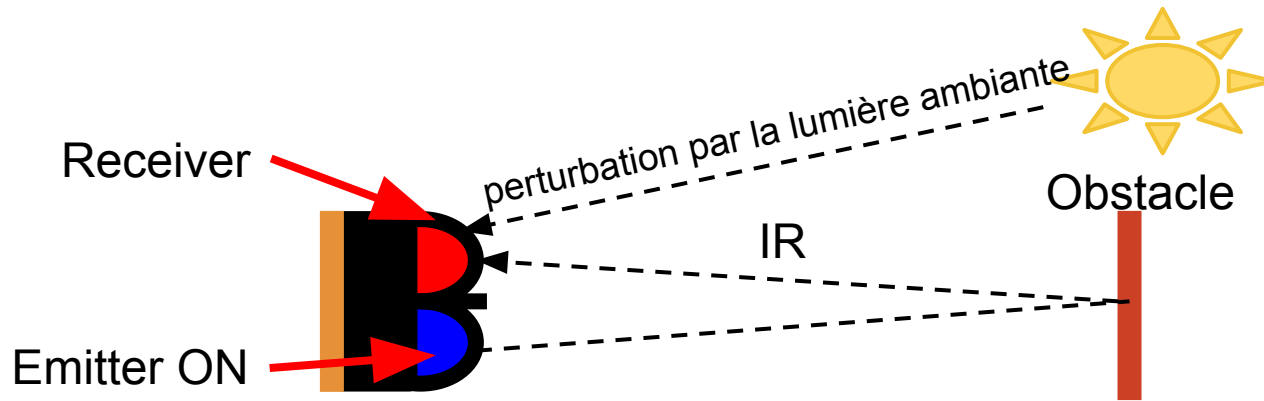


# Capteur IR - perturbations





# Capteur IR - perturbations

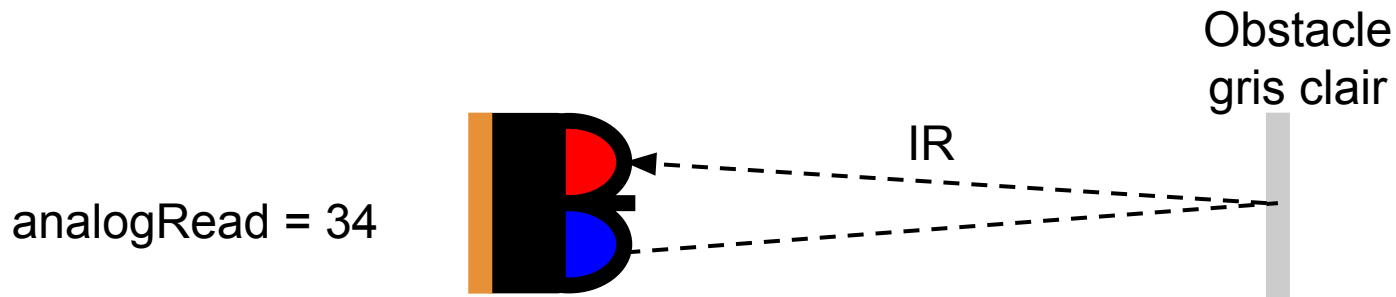
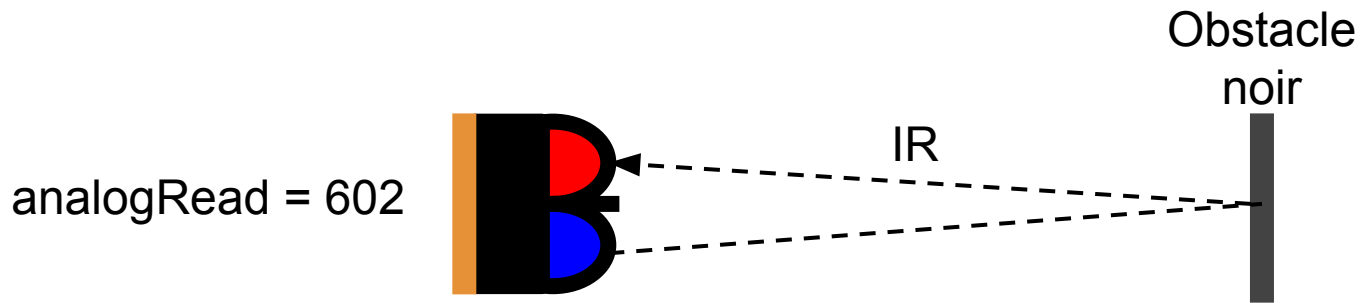


# Capteur IR - surfaces

## Vérifier les valeurs



analogRead [0,1023]



# Programmation

- Input
- DigitalRead()
- AnalogRead()
- Penser à allumer l'émetteur!

```

1  #include <prismino.h>
2
3  #define IR_SENSOR A1
4  #define IR_ENABLE 13
5
6  #define SOME_IR_CONSTANT 850
7  #define SOME_OTHER_IR_CONSTANT 400
8
9  int ir_value;
10 int ambient;
11
12 void setup() {
13     pinMode(IR_SENSOR, INPUT);
14     pinMode(IR_ENABLE, OUTPUT);
15 }
16
17 void loop() {
18     ambient = analogRead(IR_SENSOR); // ambient perturbation
19     digitalWrite(IR_ENABLE, HIGH);
20     delay(1);
21     ir_value = analogRead(IR_SENSOR);
22     digitalWrite(IR_ENABLE, LOW);
23
24     ir_value -= ambient; // subtract perturbation
25     if(ir_value > SOME_IR_CONSTANT)
26     {
27         // do something here
28     }
29     else if(ir_value > SOME_OTHER_IR_CONSTANT)
30     {
31         // do something else
32     }
33     else
34     {
35         // something else
36     }
37
38 }

```

# Utilisation

- Détecter un obstacle
- Identifier un objet
  - Attention, un IR ne voit pas comme nous!
- Lire une distance (cartographiez votre maison!)
- ...

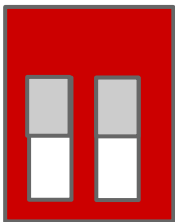
**Show time!**

# DIP switch

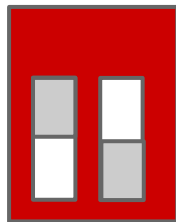
$$2^2=4$$

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

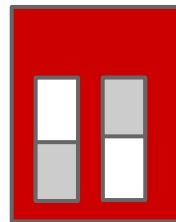
On a 2 switch avec 2 positions, donc on peut au maximum communiquer  $2^2$  positions soit 4 !  
On peut donc soit les utiliser "un par un", ce qui fait 2 boutons, soit comme un nombre binaire !



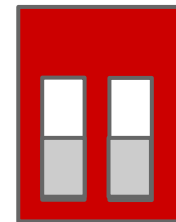
00 -> 0



01 -> 1



10 -> 2



11 -> 3

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

binaire : on "écrit" un nombre binaire qui sera entre 0 et 3

# Agenda

- Prochain démon : Utilisation des machines et réalisation d'une base mécanique

## Grand Concours

Les règles du Grand Concours  
sont disponibles

[cut.epfl.ch/concours-2015-2016](http://cut.epfl.ch/concours-2015-2016)



**FIN**

Questions?