



robotly

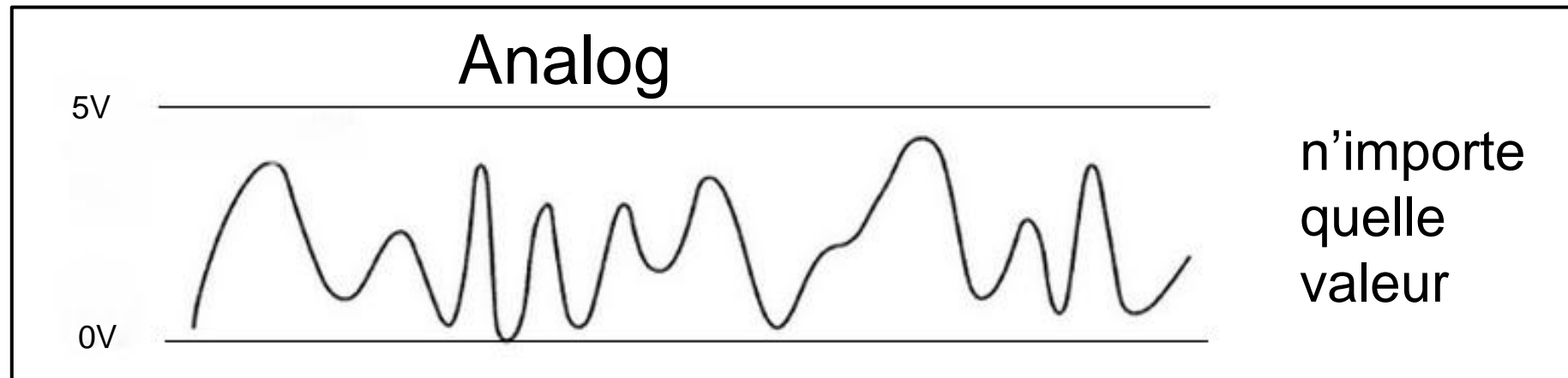
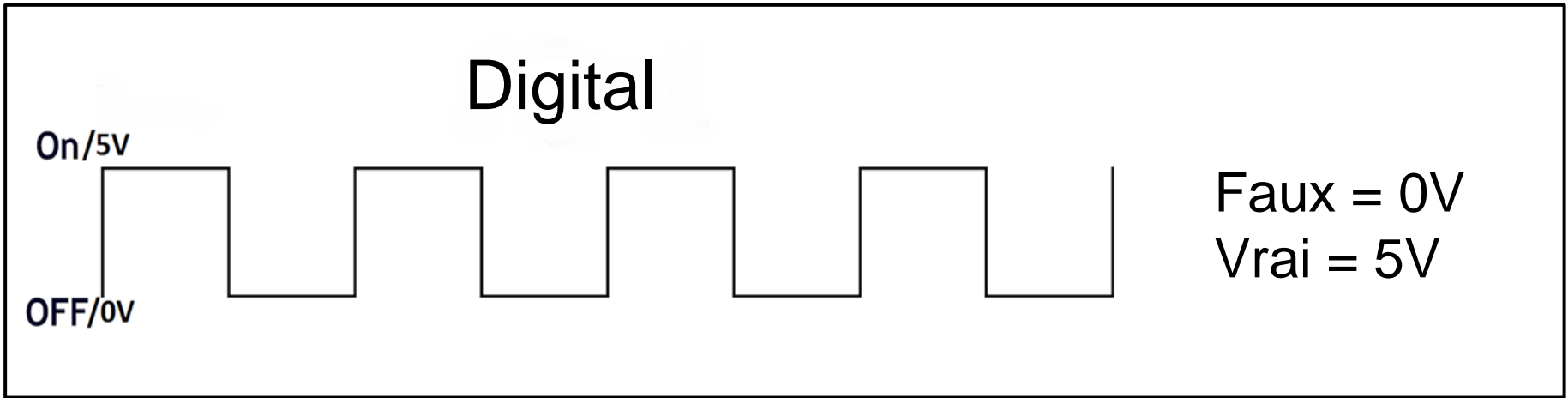
Boutons et capteurs IR

Comment voir et sentir le monde qui nous entoure

Programme du jour

- Monde digital / analogique (Rappel)
- Boutons / DIP Switch
- Capteurs IR

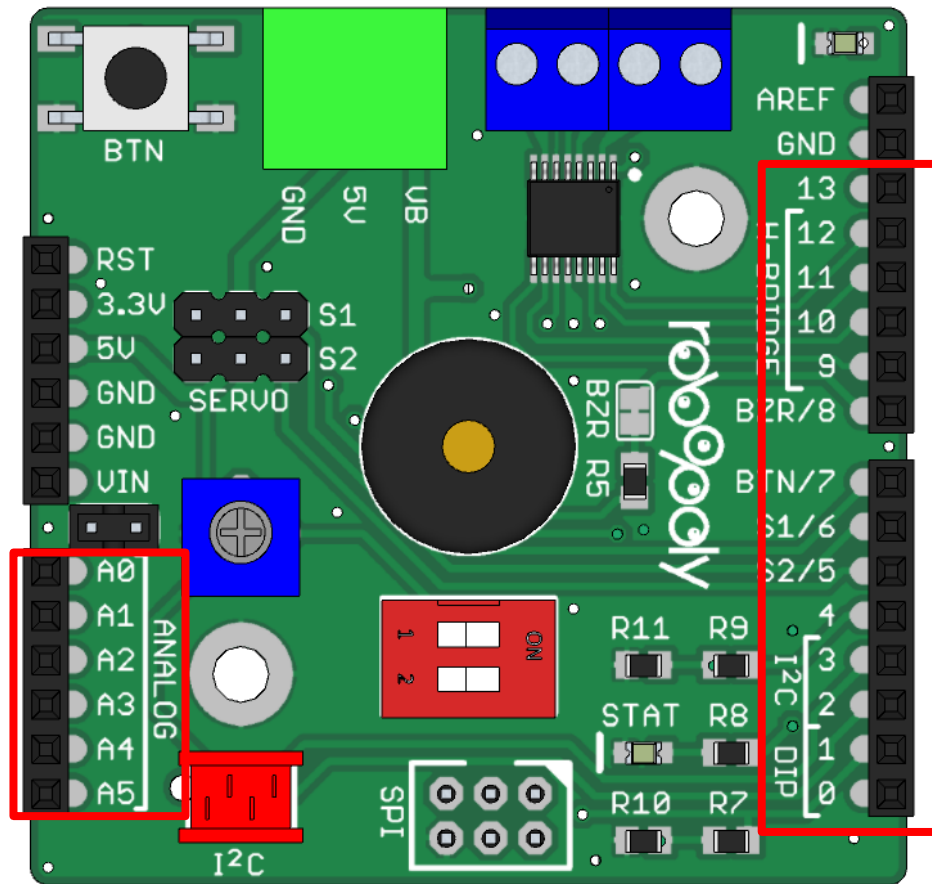
Signal digital/analogique



Pin digital/analogique

**Pins
“analogiques”
(A1-A5)**

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



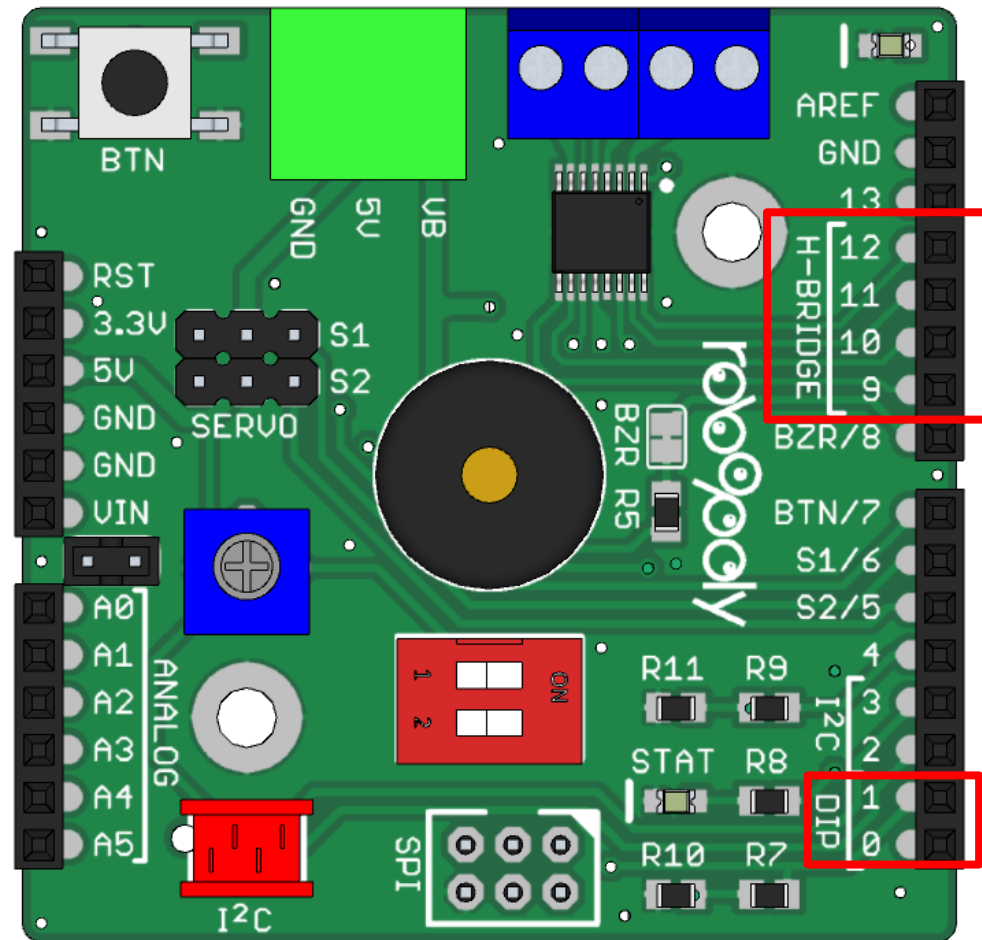
**Pins
“digitaux”
(0-13)**

Avant de faire des bêtises

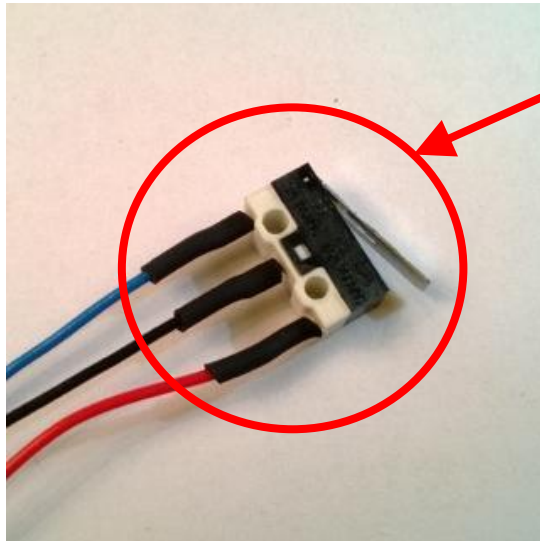
Vérifier que le pin soit libre !

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

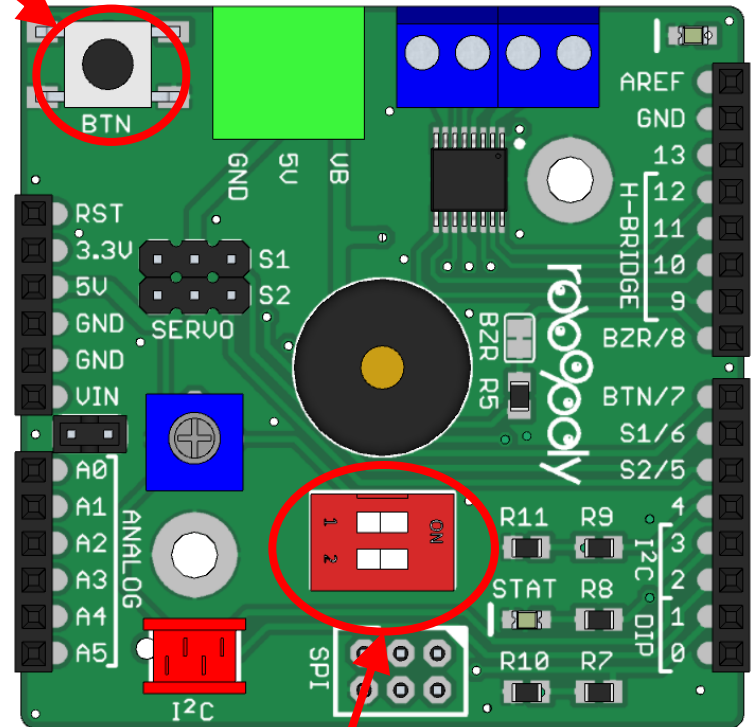
Voir le [site](#) pour savoir lesquels utiliser !



Boutons / DIP Switch



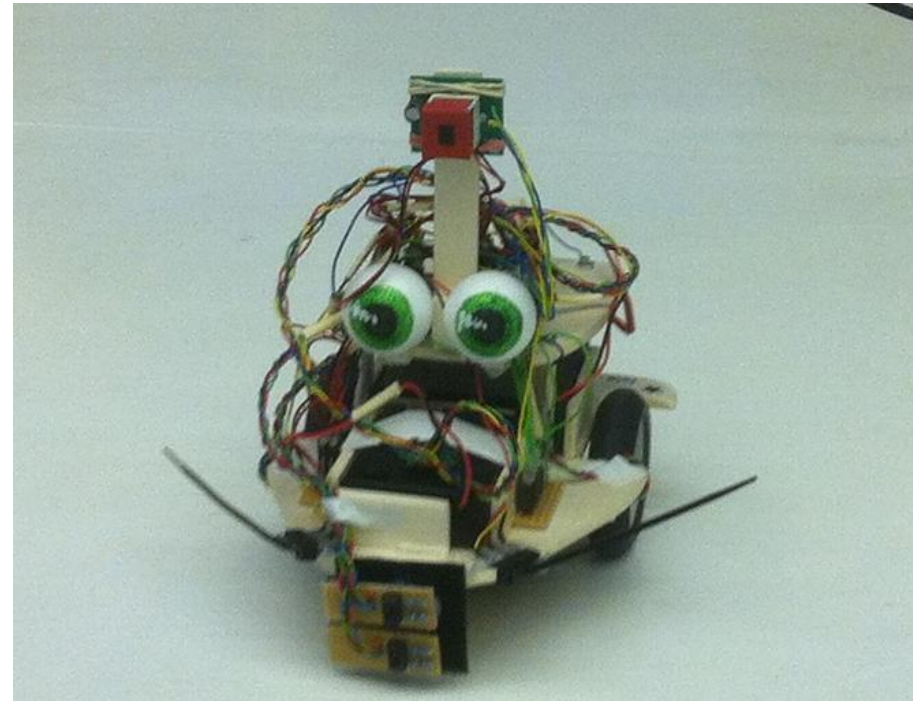
Bouton



DIP-Switch

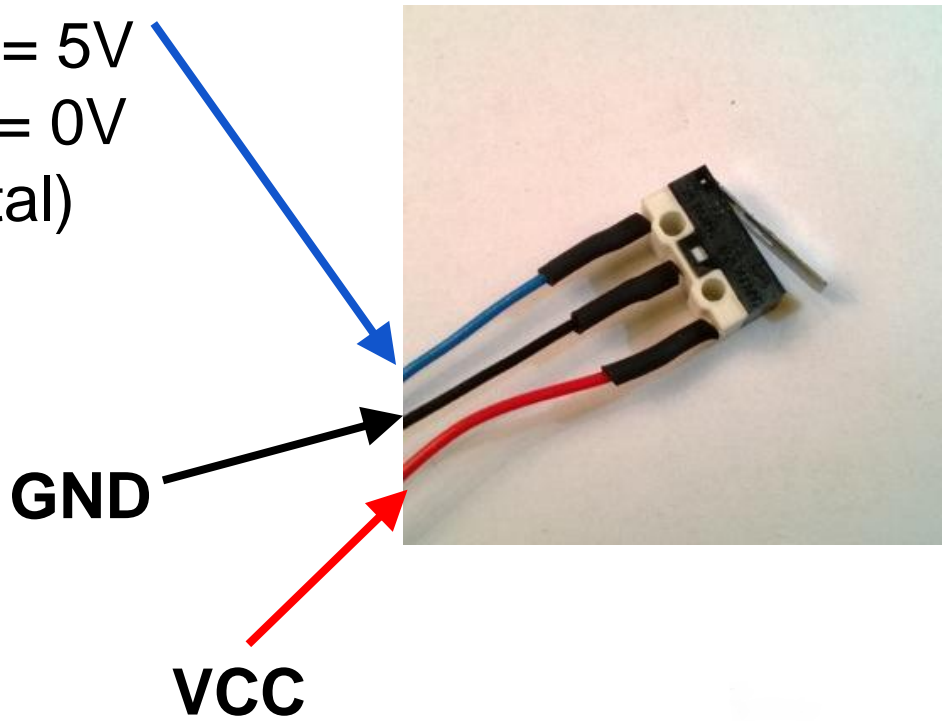
Utilisation

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



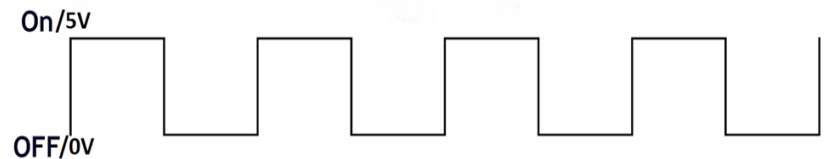
Fonctionnement du bouton

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



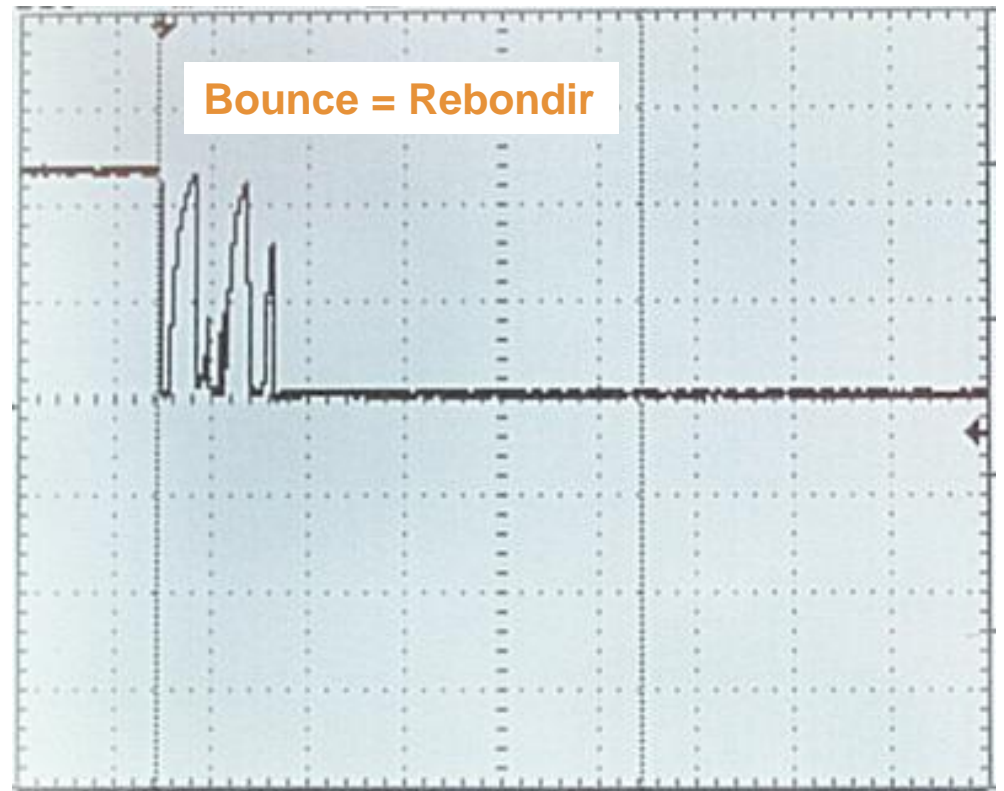
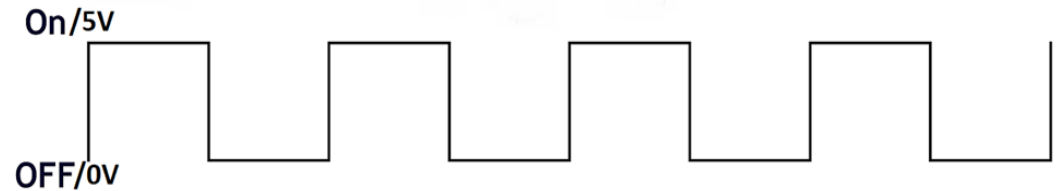
Attention rappel!

Lorsque le bouton est au 'repos' (non pressé) la valeur digitale lue est 1.



Debounce

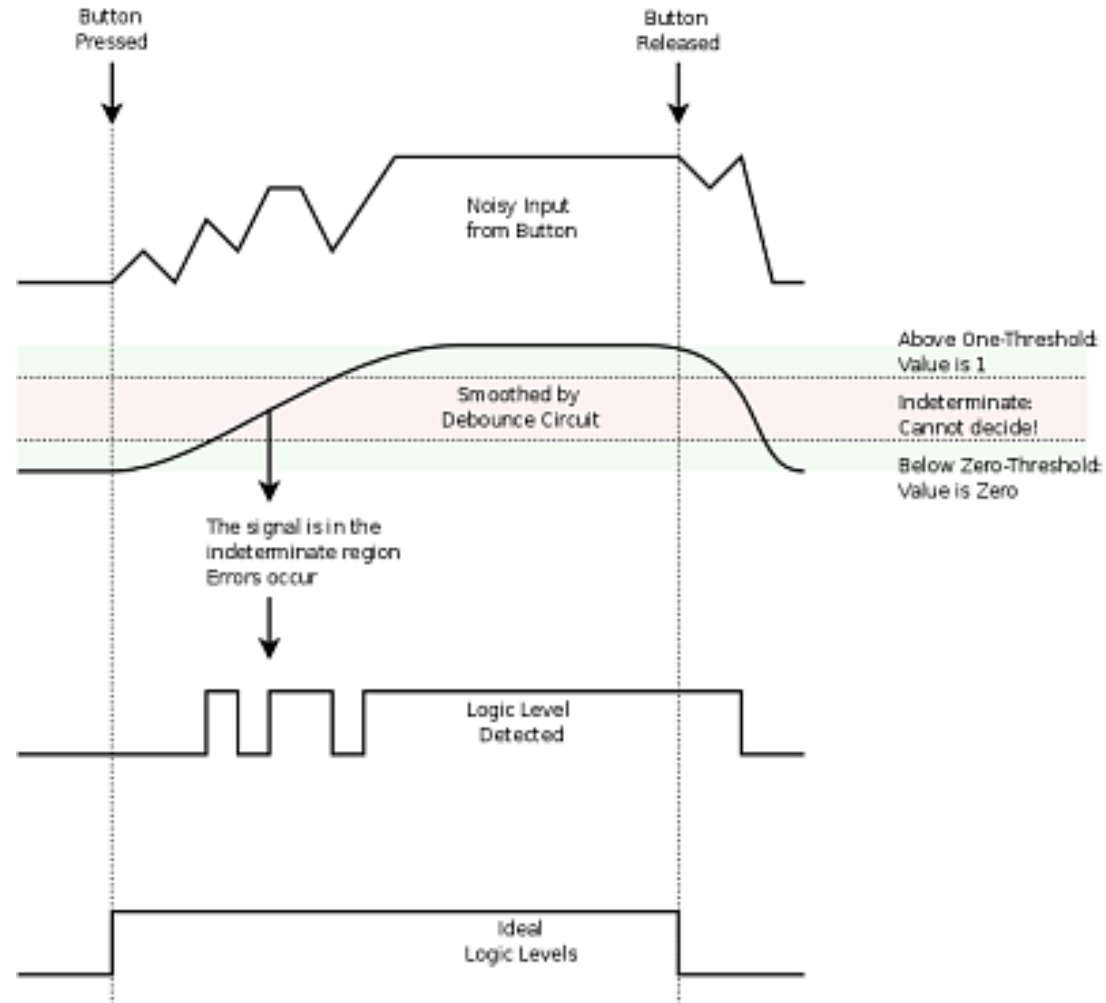
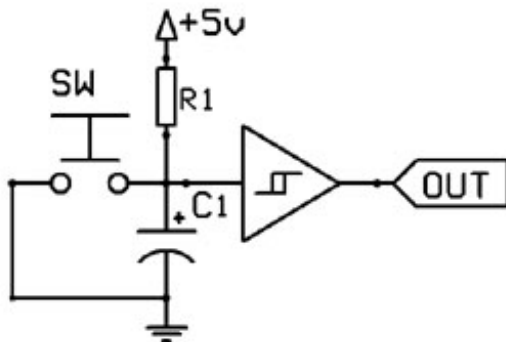
Quand on appuie sur un bouton, la mécanique cause des oscillations dans la tension lue, vous risquez donc d'avoir un comportement incertain!



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

Librairie Arduino en détail: [Librairie Arduino](#)

```

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 }

```

DIP switch

$$2^2 = 4$$

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

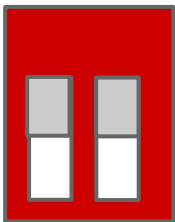
On a 2 switch avec 2 positions, donc on peut au maximum communiquer 4 (2^2) positions !
On peut donc soit les utiliser "un par un", ce qui fait 2 boutons, soit comme un nombre binaire !

Attention !

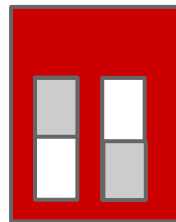
Lorsque vous utilisez le DIP switch, vous ne pouvez plus utiliser les pins digitaux 0 et 1.

Remarque !

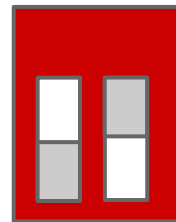
Si vous utilisez le DIP Switch, vous ne pouvez pas utiliser le module Bluetooth et inversement.



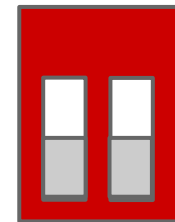
00 -> 0



01 -> 1



10 -> 2

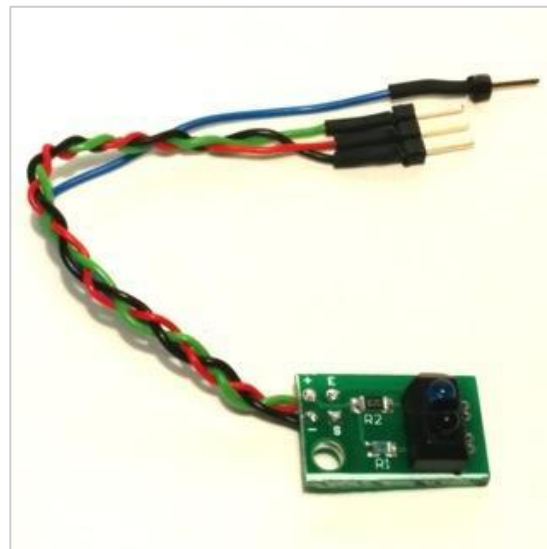
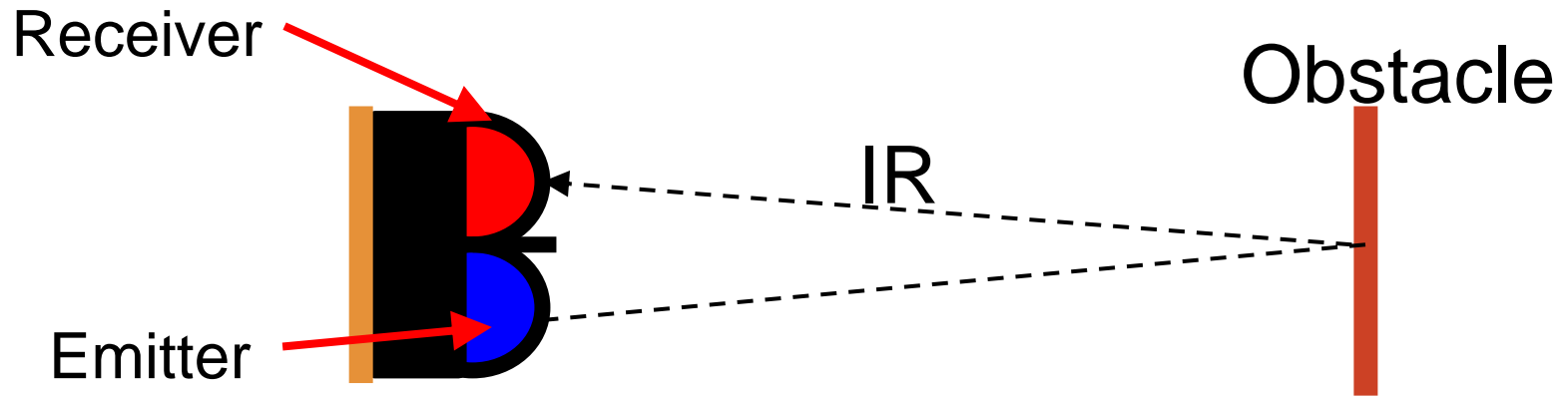


11 -> 3

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

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

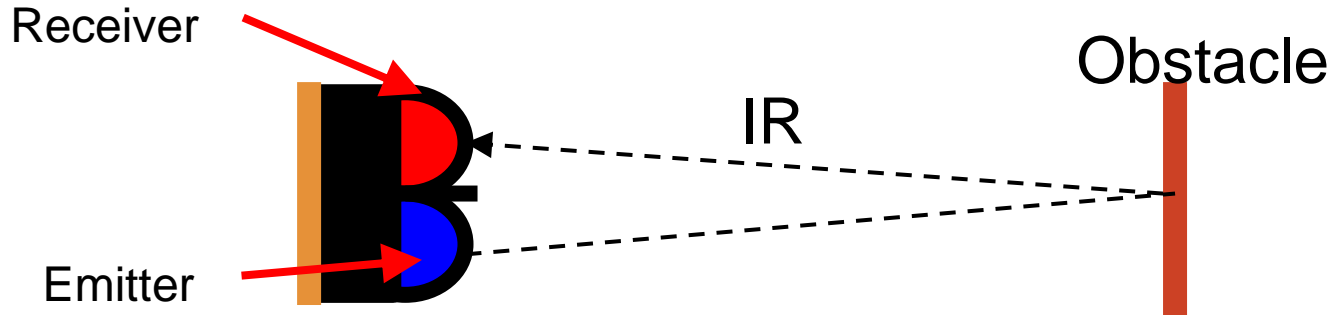
Capteur IR (infrarouge)



Utilisation

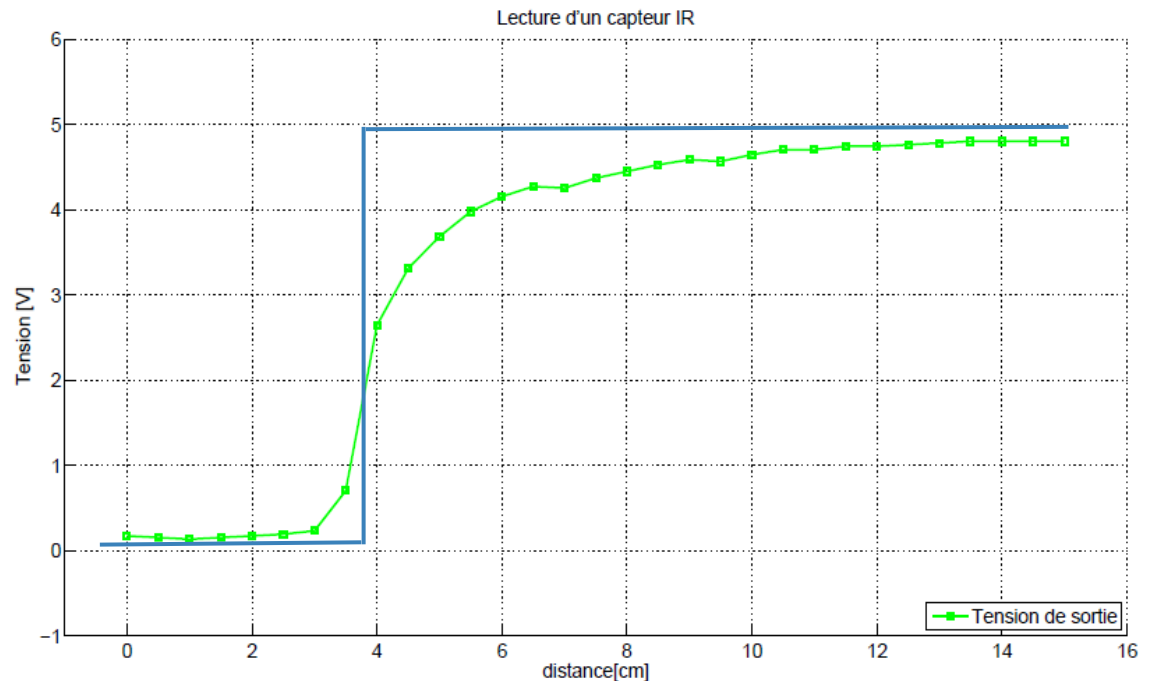
- Détecter un obstacle
- Identifier un objet
 - Attention, un IR ne voit pas comme nous!
- Lire une distance
- ...

Capteur IR - fonctionnement



Deux modes de fonctionnement :

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

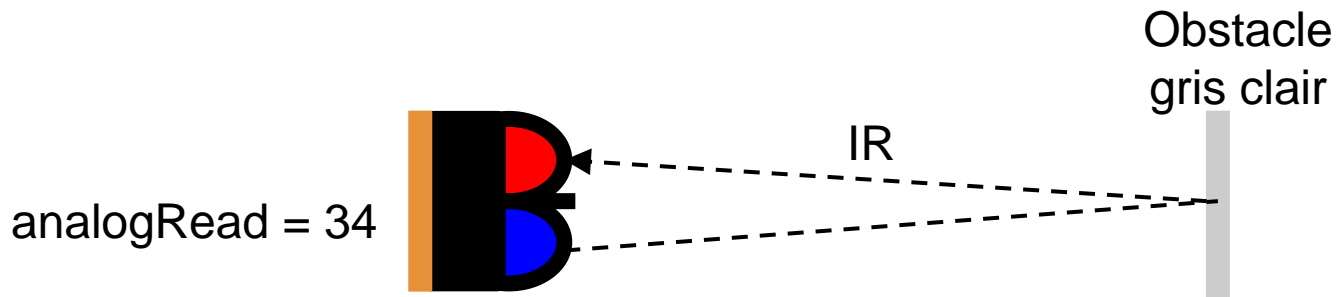
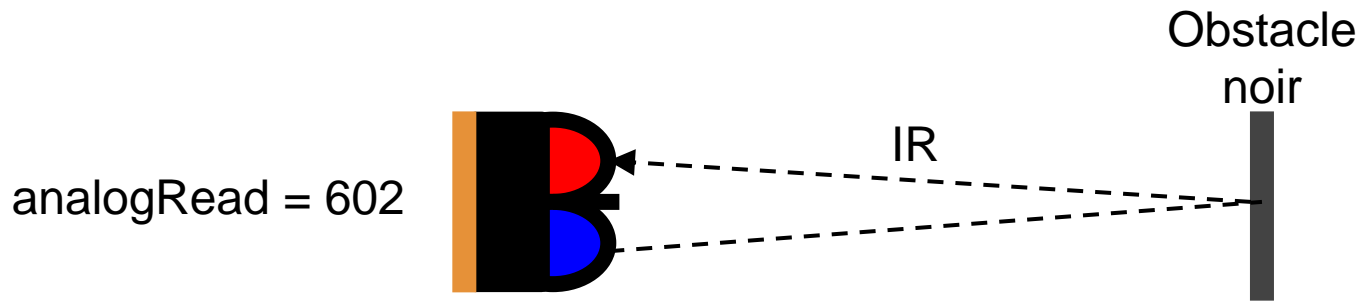


Capteur IR - surfaces

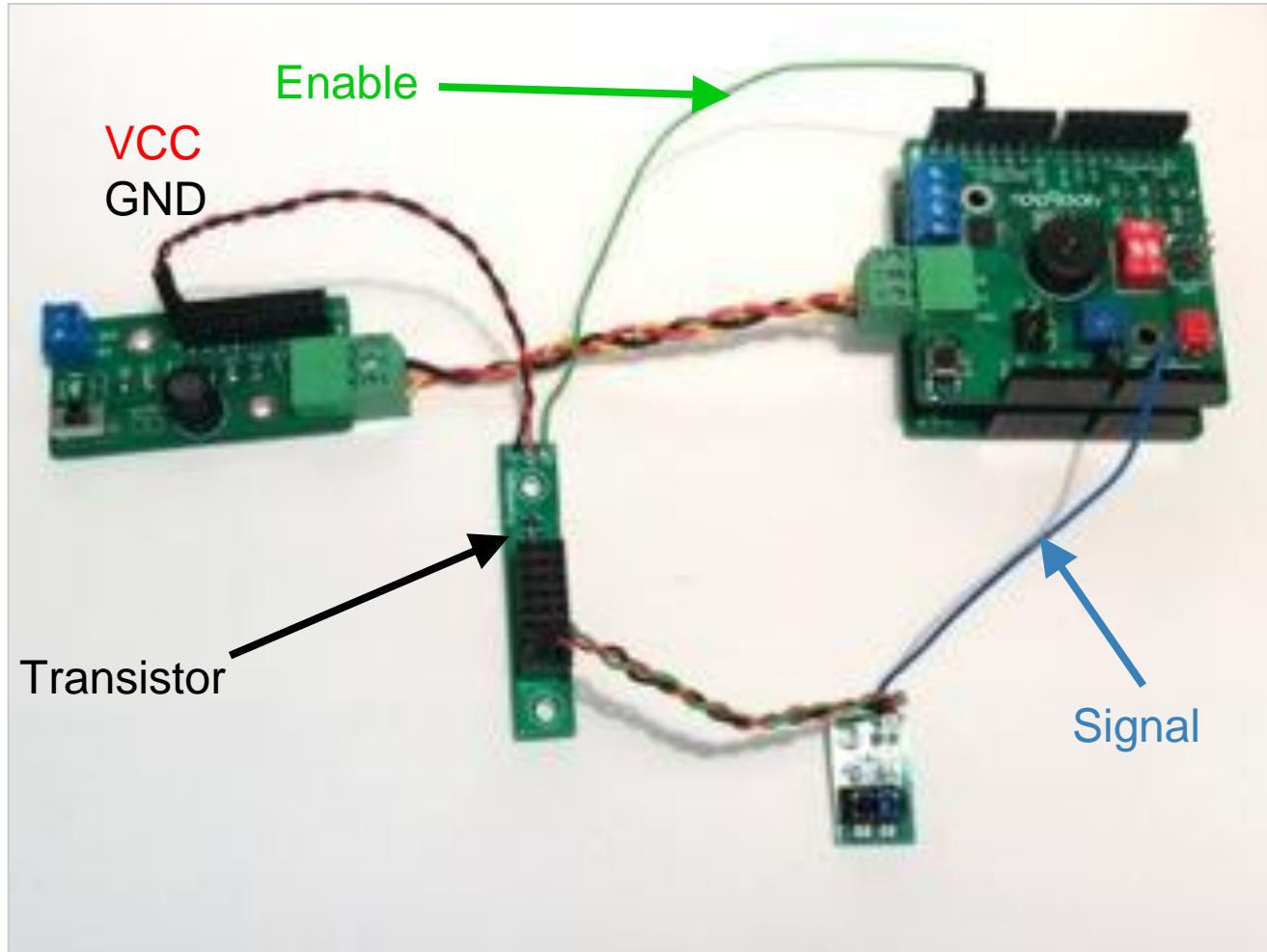


Vérifier les valeurs

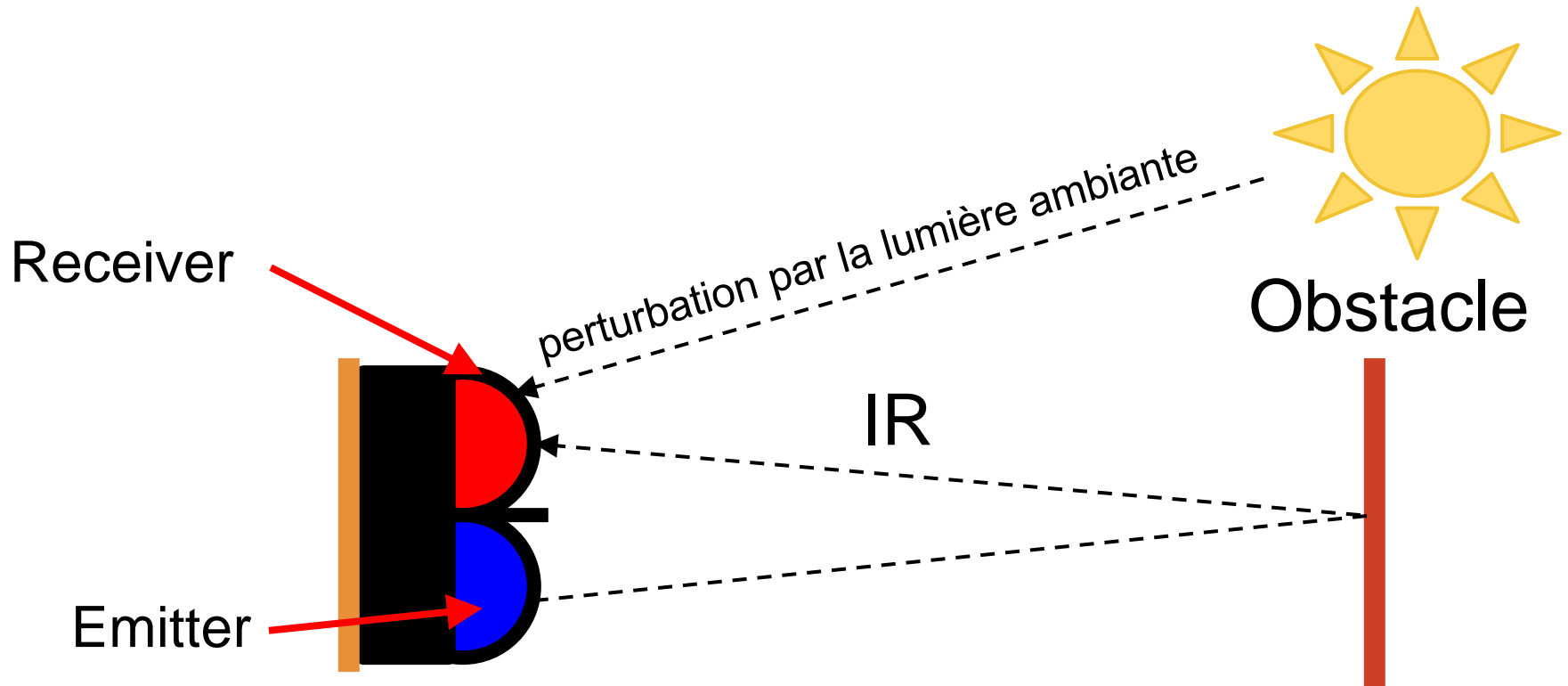
analogRead [0,1023]



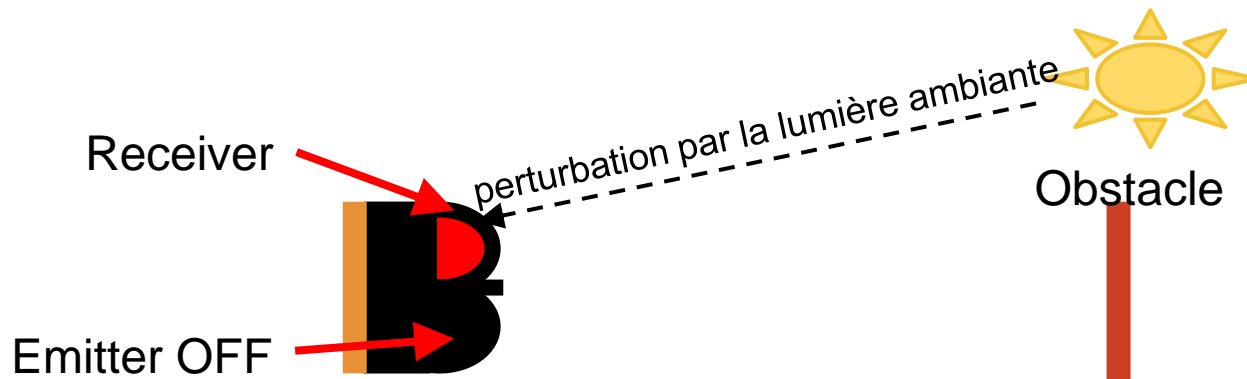
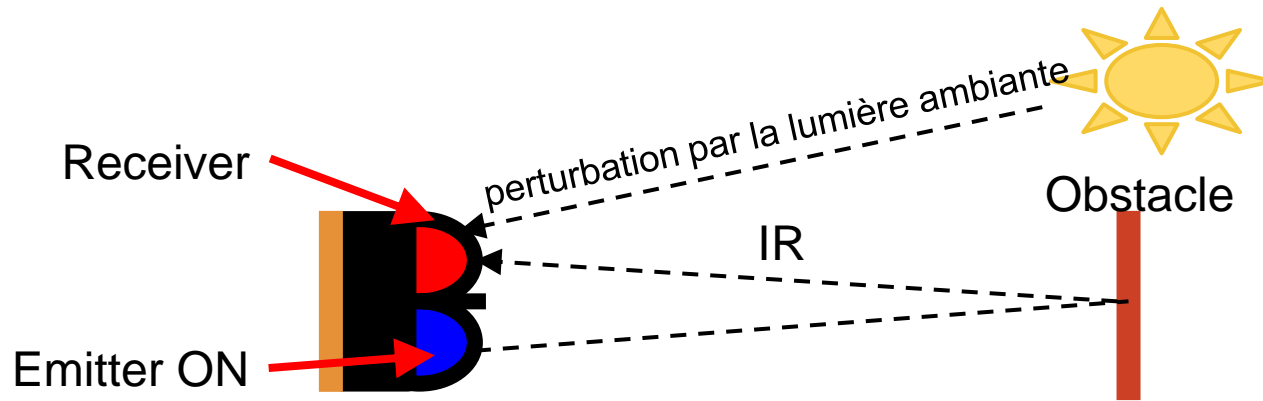
Capteur IR - branchement



Capteur IR - perturbations



Capteur IR - Gérer les perturbations



Programmation

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

```

1 #include <prismo.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 }

```

Show time!

Prochains événements

Prochain Demon

Lundi prochain, 12h15, ELA 1

Communication Serial (USB, Bluetooth)

Workshop II

Samedi 11 novembre, 9h-18h

En **haut du BM** !

Pour poursuivre le montage de ton robot,
discuter de tes projets, demander de l'aide
au comité !

Contact/Infos

Contact principal

robopoly@epfl.ch

Site officiel - toutes les infos et slides sont la!

robopoly.epfl.ch

Facebook - pour suivre l'actualité du club!

www.facebook.com/robopoly

FIN

Questions?