



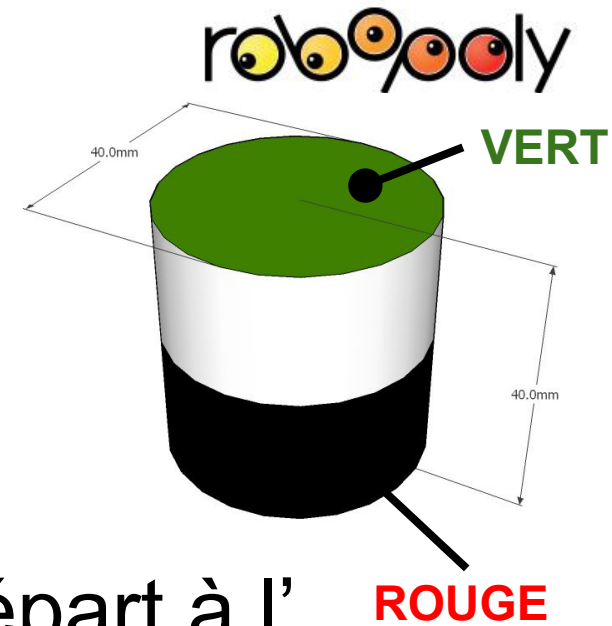
robooly

Règles du Grand Concours

Caméra linéaire

Déroulement d'un match

- Duels opposant deux robots
- Durée de 2 min
- Chacun part de sa zone de départ à l'extinction de l'émetteur IR fixé à la paroi
- But: **localiser, ramener et empiler des capsules** (petits cylindres)
- A la fin du temps, les arbitres procèdent au comptage des points et désignent le gagnant du duel

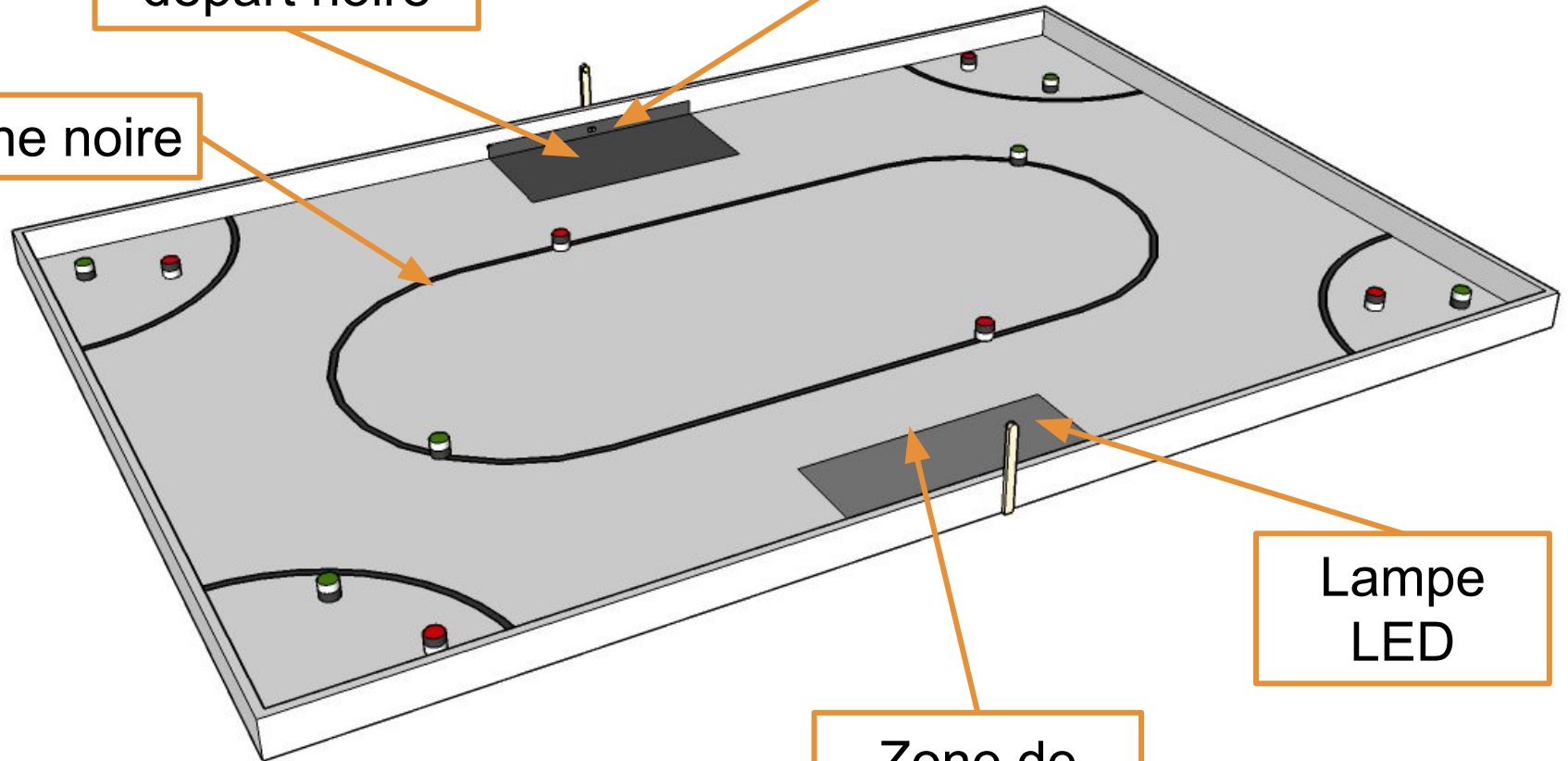


Le terrain

Zone de départ noire

Dispositif pour le départ (émetteur IR)

Ligne noire



Lampe LED

Zone de départ grise

2 m x 3 m

Comptage des points

- Réalisé à la fin du temps imparti
- Pour être considérée comme étant dans la zone, une capsule doit avoir son centre dedans et reposer directement sur le terrain

Élément validé à la fin du duel	Points gagnés
Une capsule dans la zone de départ	1 point par capsule
Deux capsules empilées	2 points par empilement
Trois capsules empilées	4 points par empilement
Quatre capsules empilées	8 points par empilement
N capsules empilées, $N \geq 5$	$8+4*(N-4)$ points par empilement
Toutes les capsules d'un empilement ont la même orientation	3 points par empilement

Exemples de comptage



- 3 capsules dans la zone
= $3 \times 1 \text{ pt} = 3 \text{ pts}$
 - 3 capsules empilées
= 4 pts
 - même orientation
= 3 pts
- Total = 10 pts

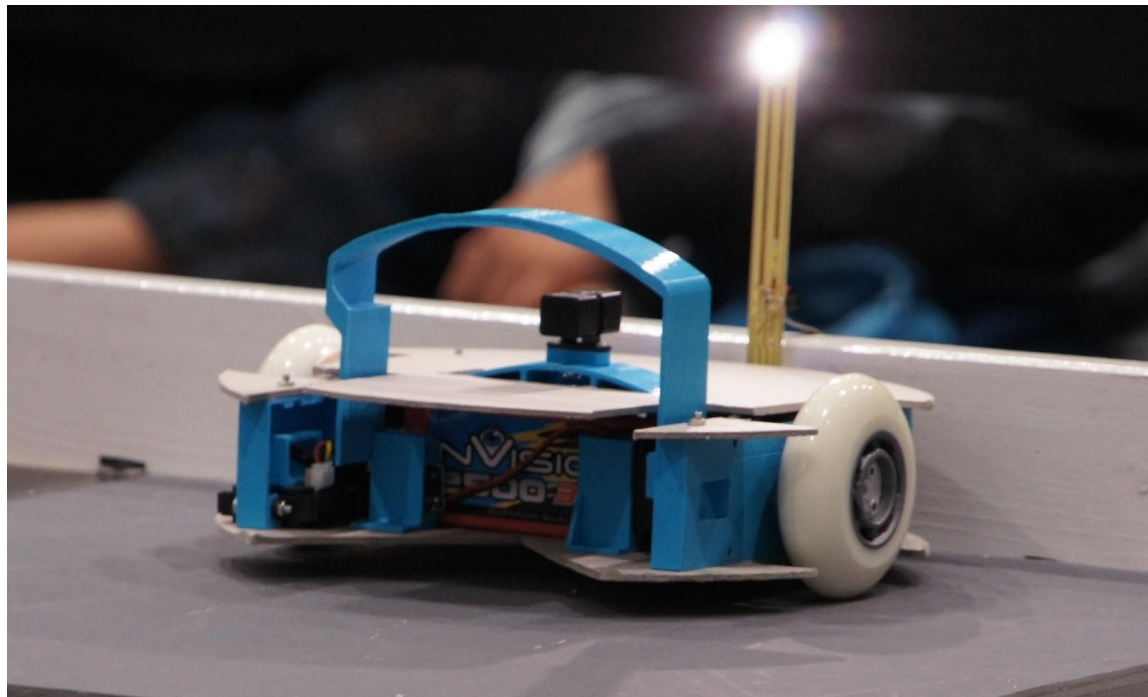
Exemples de comptage



- 2 capsules dans la zone
= $2 \times 1 \text{ pt} = 2 \text{ pts}$
 - 2 capsules empilées
= 2 pts
 - pas la même orientation
= 0 pts
- Total = 4 pts

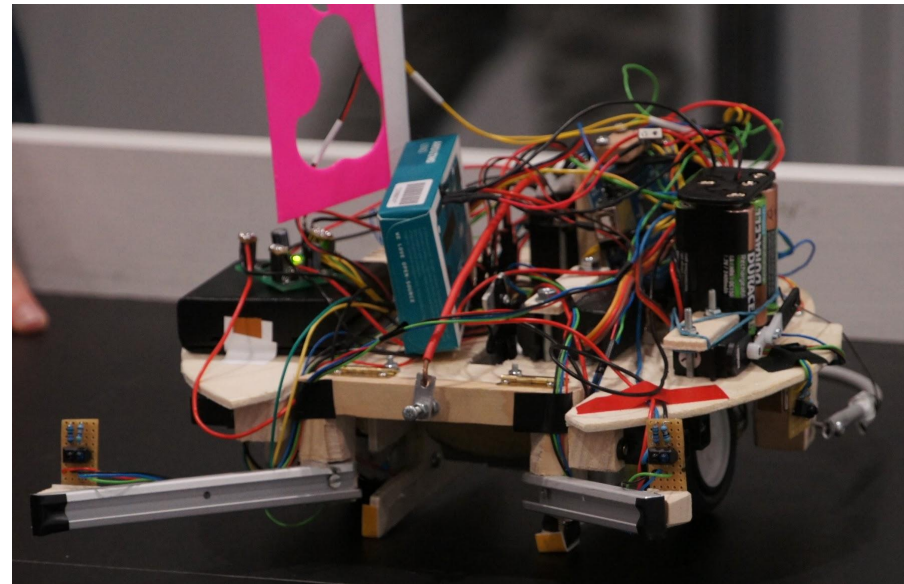
Contraintes sur le robot

- **Limite de taille** au départ du duel: cylindre de 30cm de haut et 30cm de diamètre
- Robot **autonome** sans commande humaine durant le duel ni communication sans-fil



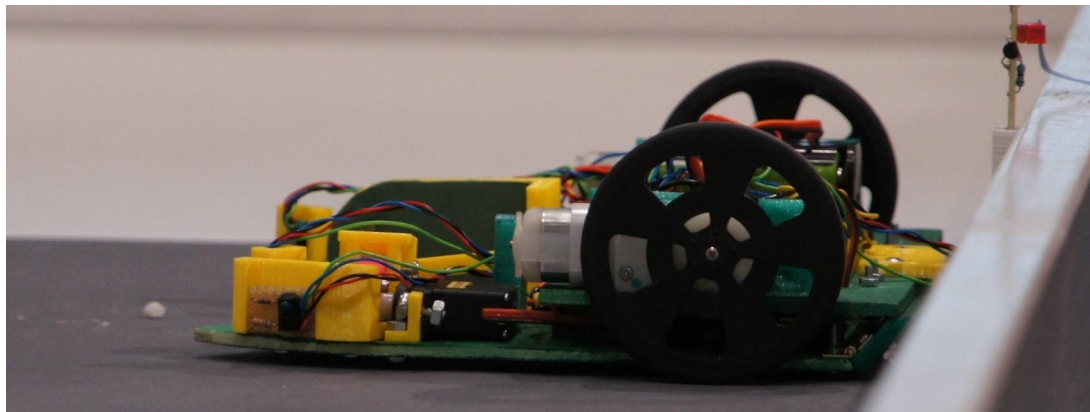
Arbitrage

- Les arbitres peuvent **intervenir** à tout moment pour séparer des robots concurrents
- Peuvent procéder, à la demande de l'équipe, à une ***intervention***: appuyer sur un bouton, et/ou ramener le robot à sa zone de départ
- Pour chaque équipe: une intervention gratuite pour tout le concours, pour la suite: coûte 1 pt



Homologation

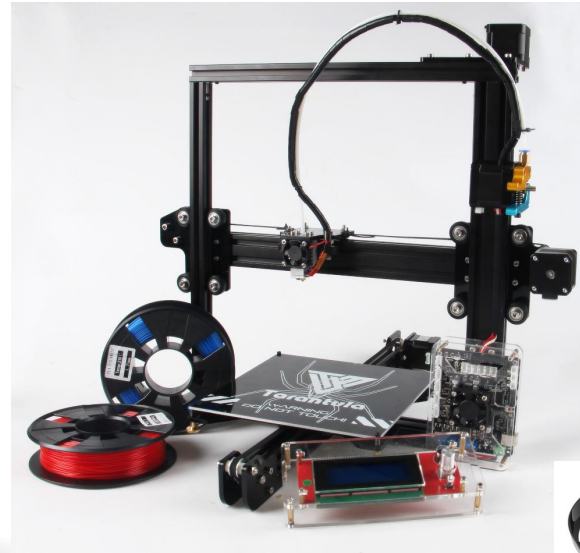
- Une fois inscrit pour participer, le robot doit passer l'homologation (possible jusqu'au jour-même)
- C'est une petite **mise en situation** permettant de s'assurer qu'il est apte à participer
- Il doit: réussir un départ automatisé avec émetteur IR et ramener à sa zone de départ une capsule (reliée à la zone par une ligne)



Prix!

- Le gagnant est celui **ayant obtenu le plus de points** à la fin des duels. Il choisit un des prix suivants:

- Imprimante 3D
- Tablette *Nvidia K1*
- Quadcopter *Parrot*



- **Le plus beau robot** (désigné par le public) recevra un prix surprise!

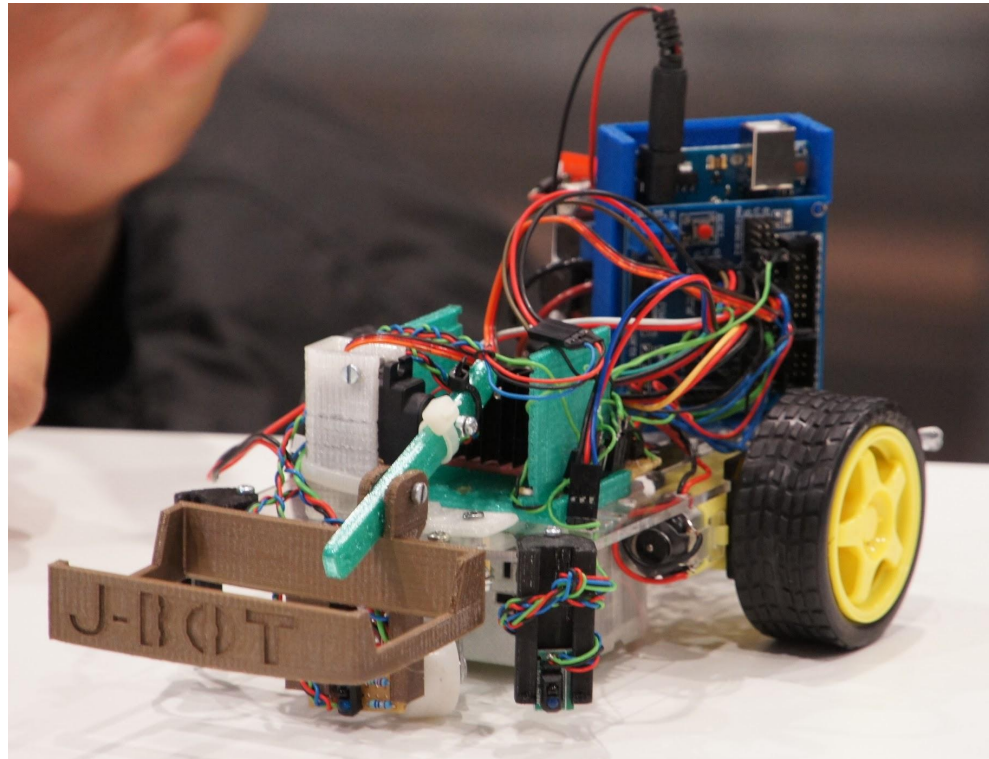
L'événement

- **Samedi 19 mars 2016**
- 10h - 14h: Préparation participants, des robots et tests en conditions
- **14h: début du concours**
- + stand du comité avec pleins de projets et démos cools...

Intéressés ?

- N'oubliez pas de **vous inscrire!**
- Il faut être membre de l'association
- Samedi 5 mars: **Workshop III**
 - Pour finaliser votre robot
 - Homologation (une intervention bonus)
- [Plus d'infos sur le site](#)
- Le terrain est déjà en place au local!

Des questions sur le Grand Concours ?



The word "robooly" is written in a large, black, rounded font. Each letter contains a stylized eye: 'r' is yellow, 'o' is yellow, 'b' is orange, 'o' is orange, 'l' is orange, 'y' is red. The background is a collage of technical elements: a printed circuit board (PCB) with various components, a blue breadboard, a red terminal block, and snippets of C++ code. The code includes comments like "ultrasonic sensor and button values" and "digitalRead(LEADER_BUTTON);", and function calls like "digitalWrite(LED, digitalRead(LEADER_BUTTON));".

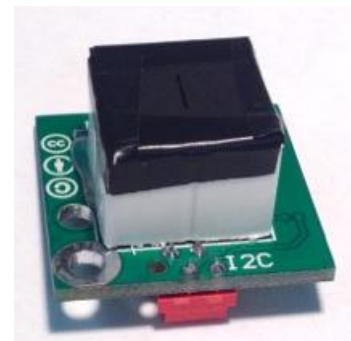
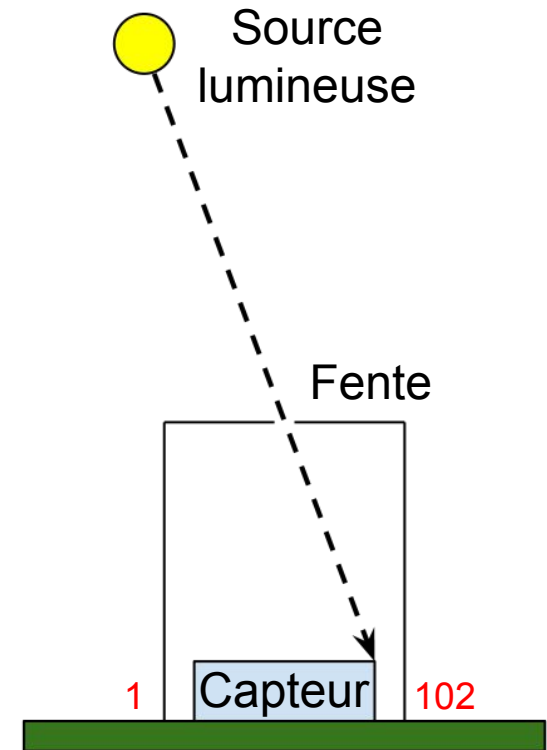
robooly

Règles du Grand Concours

Caméra linéaire

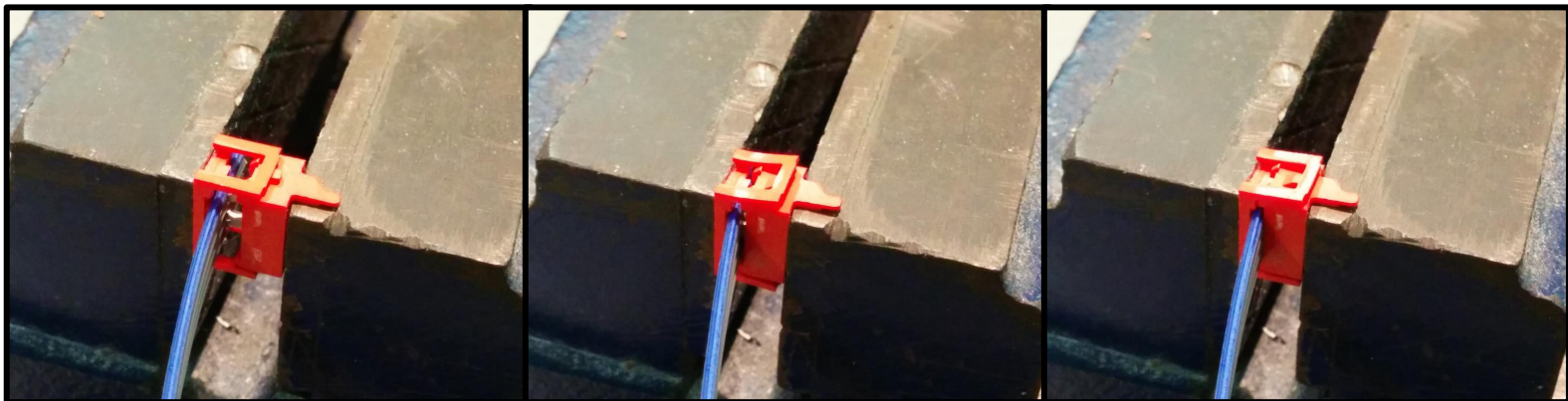
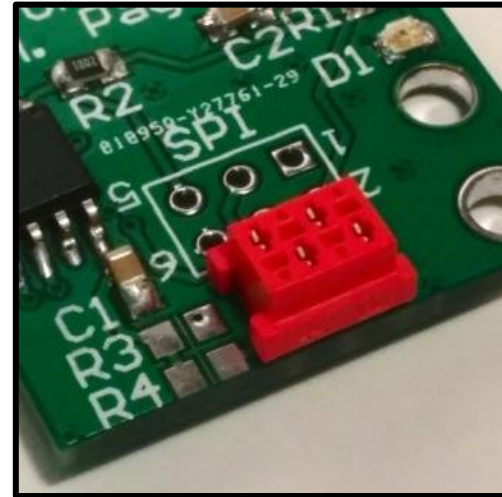
Qu'est-ce ?

- Une caméra de 102 pixels de large par 1 pixel de hauteur
- Donne la **valeur d'intensité lumineuse** entre 0 et 255 pour chacun des pixels
- Applications:
 - Localisation
 - Lecture de code barre
 - Flux optique
 - Suivi de ligne



Connection

- Connecteur de type “microMatch” sur le shield
- avec nappe à 4 fils
- Comment assembler connecteur et nappe ?
 - garder le fil bleu du côté du détrompeur
 - serrer le connecteur à l'étau



Programmation - installation

- Télécharger le firmware sur la caméra en utilisant le bootloader du local
- Télécharger la librairie LinearCamera dans Arduino IDE
- Commencer à coder!

Programmation - fonctions

- La caméra linéaire est un objet C++ (programmation orientée objet)
- Pas besoin de maîtriser le c++ pour l'utiliser
- `LinearCamera lc = LinearCamera (5)`
 - déclaration de l'objet lc (C++)
- `lc.setExpTime (time)`
 - `time` = temps d'exposition en μs
 - par défaut il est de $150 \mu\text{s}$

Programmation - fonctions

- **lc.getPixels ()**
 - lance l'acquisition d'une nouvelle image (nécessaire à chaque lecture)
 - renvoie un *pointeur* vers un tableau de **uint_8t** (= **int** compris entre 0 et 255) contenant les valeurs de pixels
- **lc.getPeak ()**
 - renvoie l'indice du pixel qui a la valeur maximale (le pic d'intensité)

Code - Récupérer le pic



```
#include <prismo.h>
```

Inclure la bibliothèque

```
#include <LinearCamera.h>
```

Définir l'objet caméra

```
LinearCamera lc = LinearCamera(5);
```

```
int lcPeak = 0;
```

Créer la variable qui va contenir le pic

```
void setup() {  
    Serial.begin(9600);  
}
```

Récupérer les pixels de la caméra (à chaque passage)

```
void loop() {  
    lc.getPixels();  
    lcPeak = lc.getPeak();  
    Serial.println(lcPeak);  
    delay(100);  
}
```

Chercher l'indice du pic parmi les pixels

Afficher l'indice

Code - Parcourir les pixels



```
#include <prismo.h>
```

```
#include <LinearCamera.h>
```

```
LinearCamera lc = LinearCamera(5);
```

```
uint8_t *pixels = NULL;
```

```
void setup() {  
    Serial.begin(9600);  
}
```

```
void loop() {  
    pixels = lc.getPixels();  
    for(int i = 0; i < LinearCamera::Pixels; i++)  
        Serial.print(pixels[i])  
    Serial.print("\n");  
    delay(500);  
}
```

Inclure la bibliothèque

Définir l'objet caméra

Créer le pointeur vers les pixels

Récupérer les pixels de la caméra (à chaque passage)

Macro vers le nb. de pixels de la camera

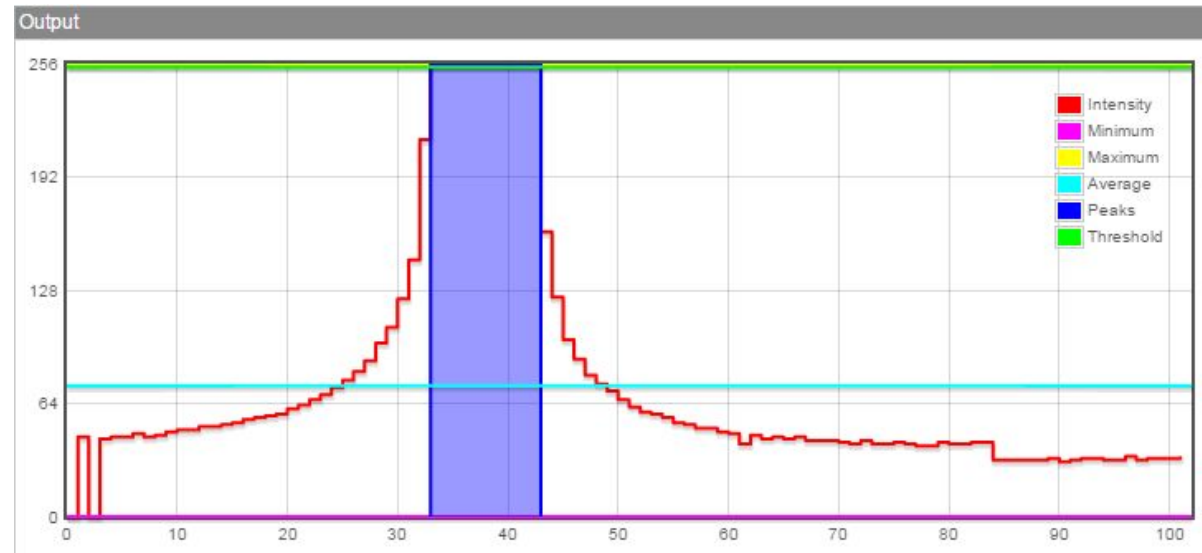
Afficher les pixels

Le Robopoly Control Center



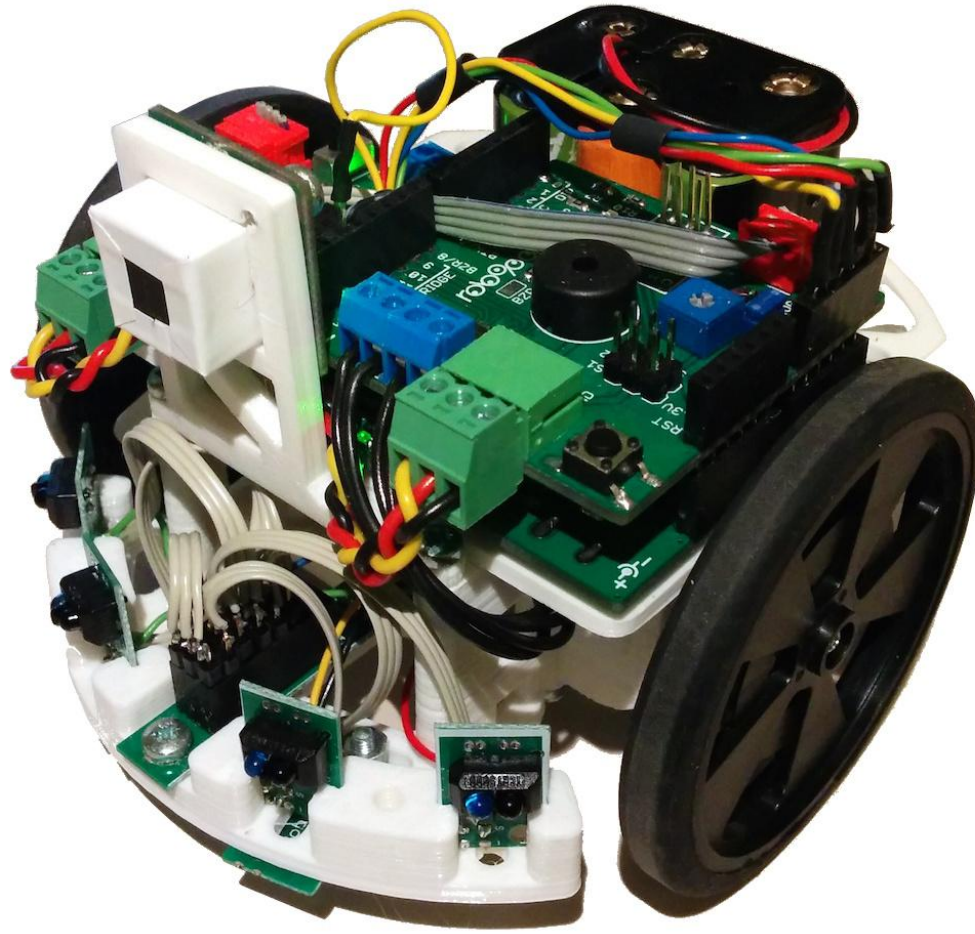
- Application Chrome pour visualiser les capteurs du kit PRisme sur une interface graphique
- Très utile pour visualiser les pixels de la caméra linéaire ou les capteurs infra-rouges

- Téléchargeable depuis le [Chrome Web Store](#)



- Nécessite le téléchargement d'un [programme spécial](#) sur le PRismo

Démonstration!



Aller plus loin...

- Si le temps d'exposition est trop élevé, beaucoup de pixels vont saturer à 255
 - → ajuster dynamiquement le temps d'exposition
- D'autres challenges:
 - Trouver la position de plusieurs pics à la fois
 - Utiliser la caméra pour suivre une ligne
 - Utiliser plusieurs caméras en même temps
 - Estimer une vitesse en implémentant un "optical flow" (en.wikipedia.org/wiki/Optical_flow)

Nouveau ?

- Inscription toujours possible
- Venir nous voir à la fin de la présentation ou nous envoyer un mail
- 20 .-
- Permet un accès 24h/24 7j/7 au local jusqu'en août prochain
- Présentations du premier semestre:
robopoly.epfl.ch/demons

Calendrier du semestre

- **Prochain démon**
 - Lundi 29 février , 12h15 en ELA1
 - Sujet: Timers et interruptions
- **Workshop III**
 - Samedi 5 mars en haut du BM
 - Utile pour se préparer au concours
- **Grand Concours**
 - Samedi 19 mars en SG1

Contact/Infos

Contact principal

robopoly@epfl.ch

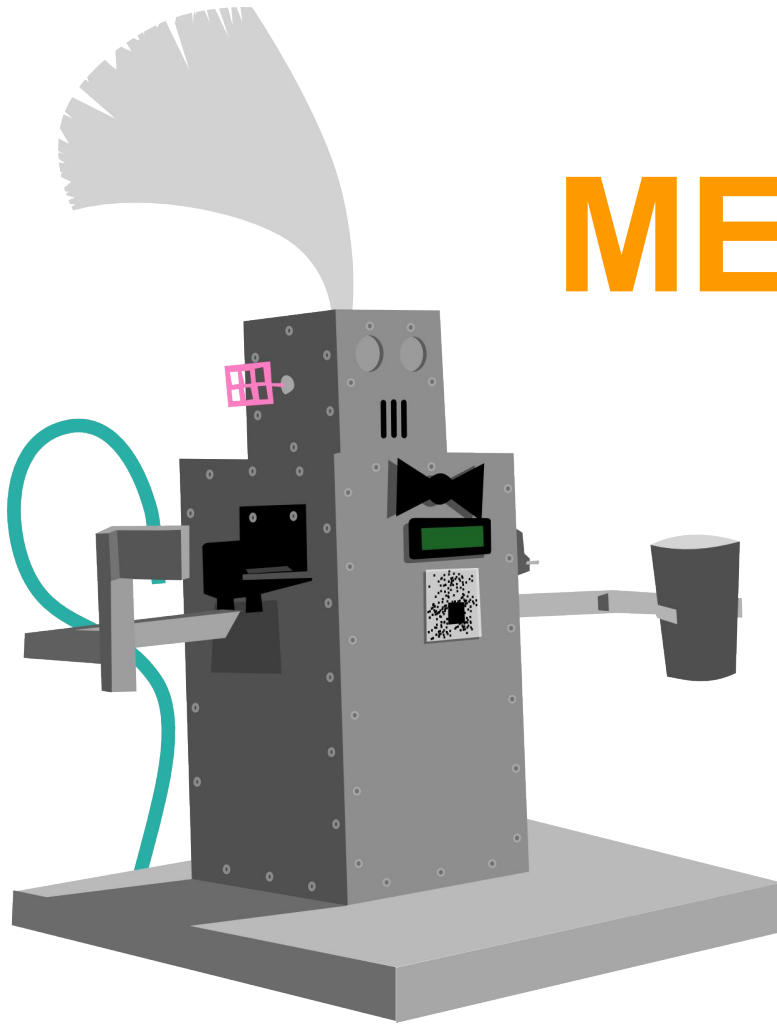
Site officiel - toutes les infos et slides sont la!

robopoly.epfl.ch

Facebook - pour suivre l'actualité du club!

www.facebook.com/robopoly

MERCI!



Questions?