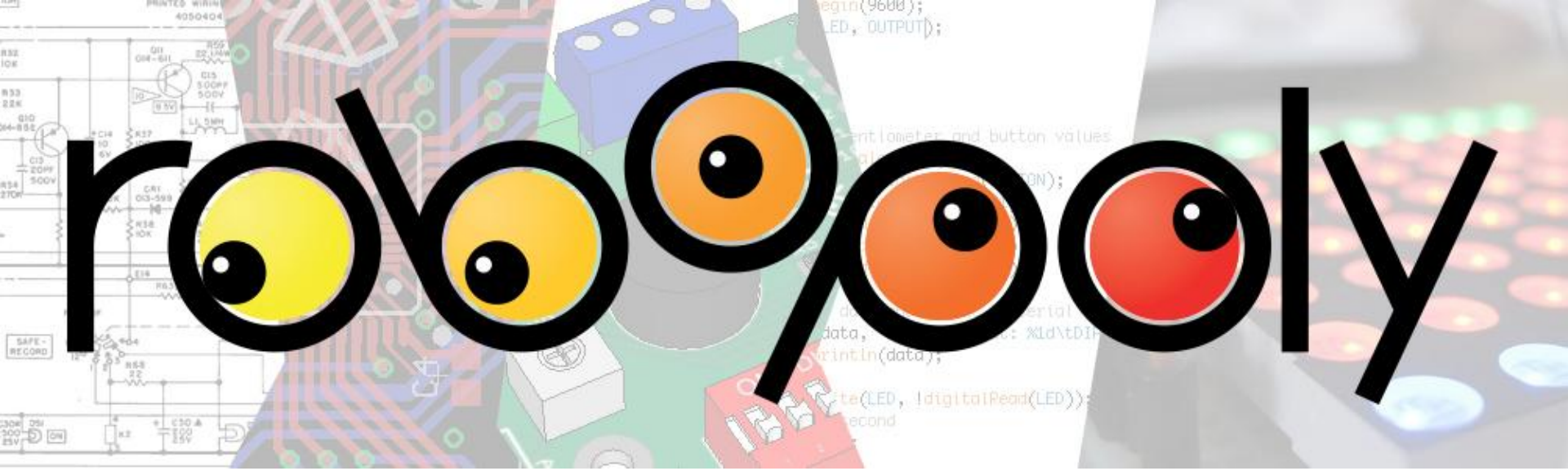


Les encodeurs

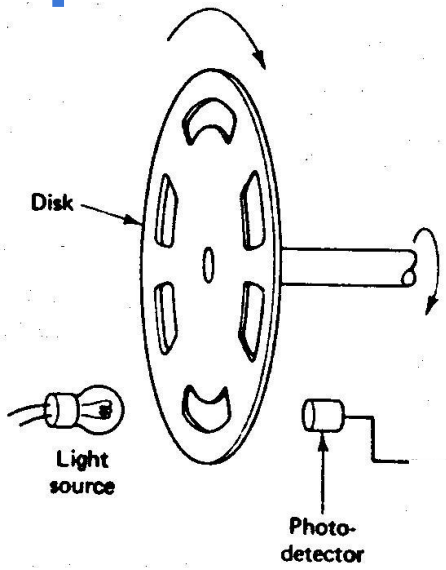
Problématique générale

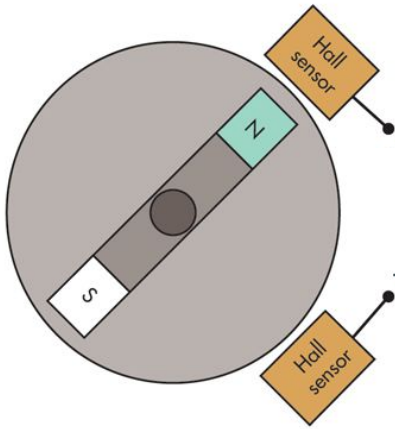
- Comment **se localiser dans un environnement** inconnu ou en mouvement ?
  - Pour savoir revenir à son point de départ
  - Pour cartographier un lieu
  - Pour aller d'un point connu à un autre
- Comment **déterminer les mouvement réels** du robot ?
  - Pour adopter une vitesse précise
  - Pour détecter certains événements
- Les **encodeurs** peuvent être **une solution** (mais il en existe d'autres)

Qu'est-ce que c'est ?

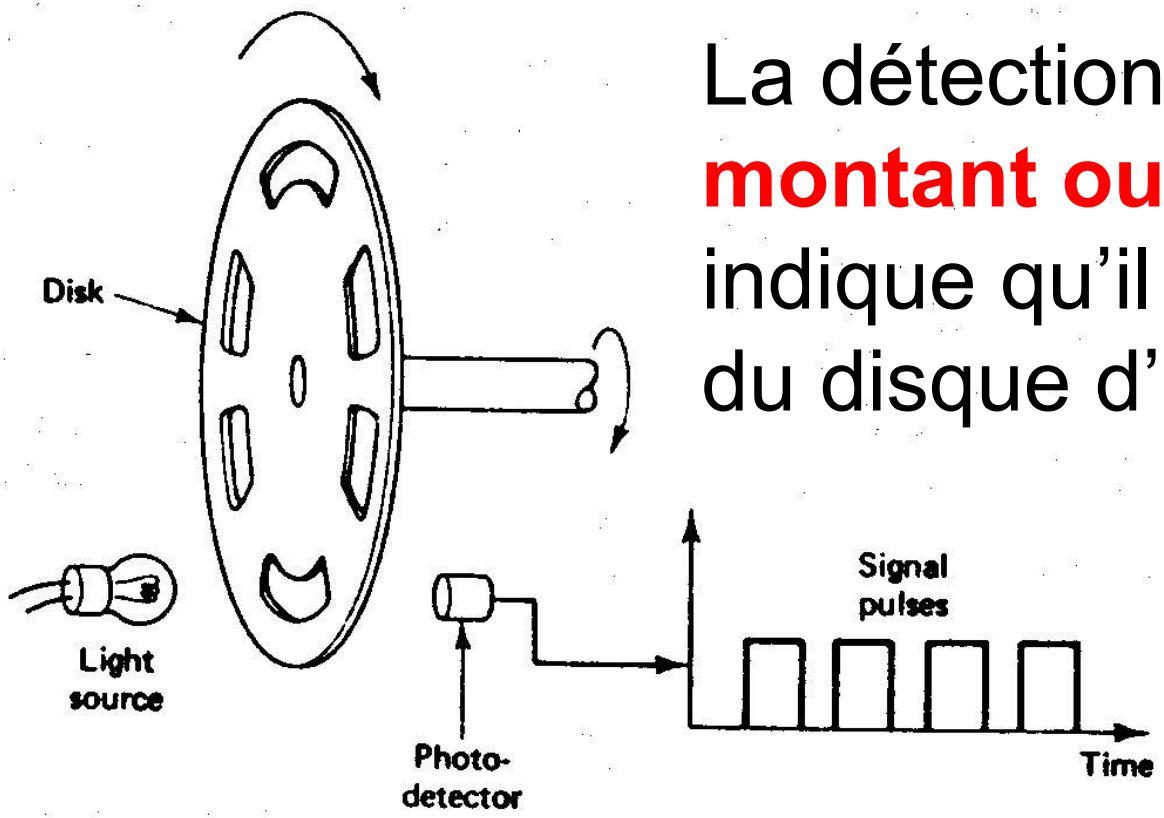
- Un système permettant de connaître à tout moment et de **manière précise** l'**angle de rotation de la roue**

- Différents types:
  - **optique**
    - capteurs IR ou photorésistances
  - **magnétique**
    - aimants et capteurs à effets Hall





Fonctionnement général

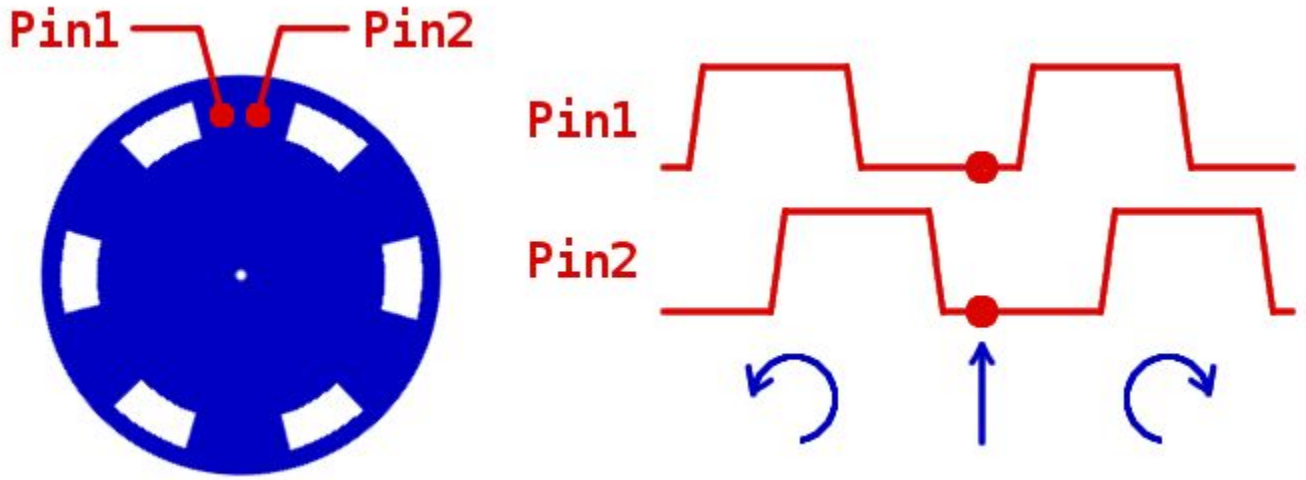


La détection d'un **flanc montant ou descendant** indique qu'il y a eu rotation du disque d'un certain angle

- Le nombre de trous dans le disque donne l'**angle de résolution** de l'encodeur

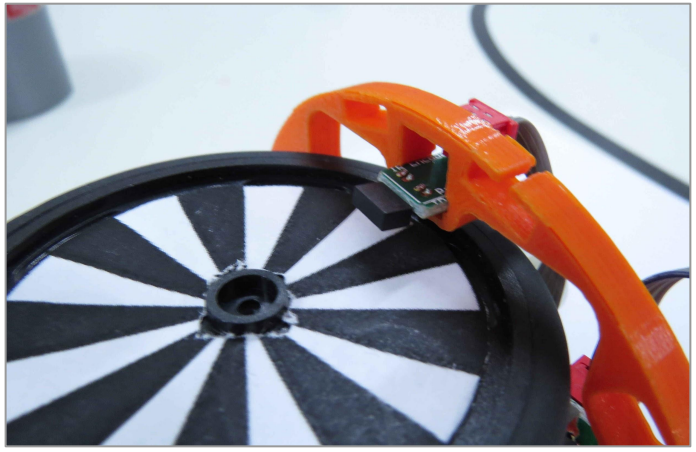
Fonctionnement général

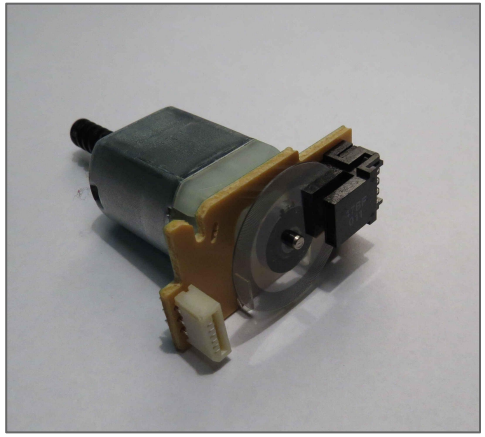
- **Deux capteurs** en **quadrature** permettent de connaître le **sens de rotation** et de quadrupler la résolution



Encodeur optique

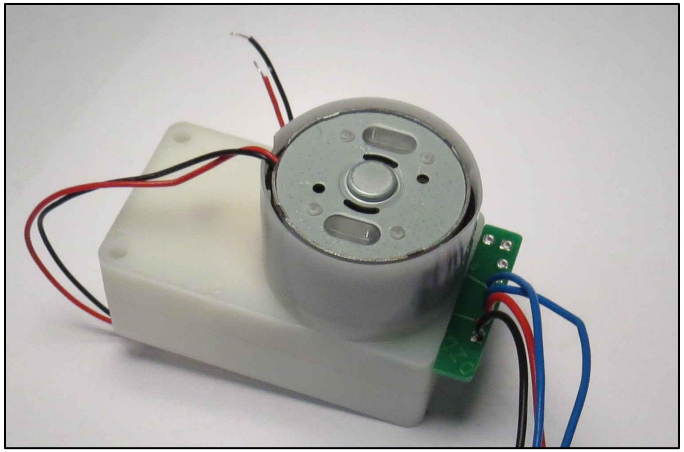
- Utiliser des capteurs infrarouges
- Système à fixer directement sur la roue
  - Motif en papier **noir et blanc**
  - Utiliser un pattern **déjà moulé** dans la roue (celles du kit en ont!)

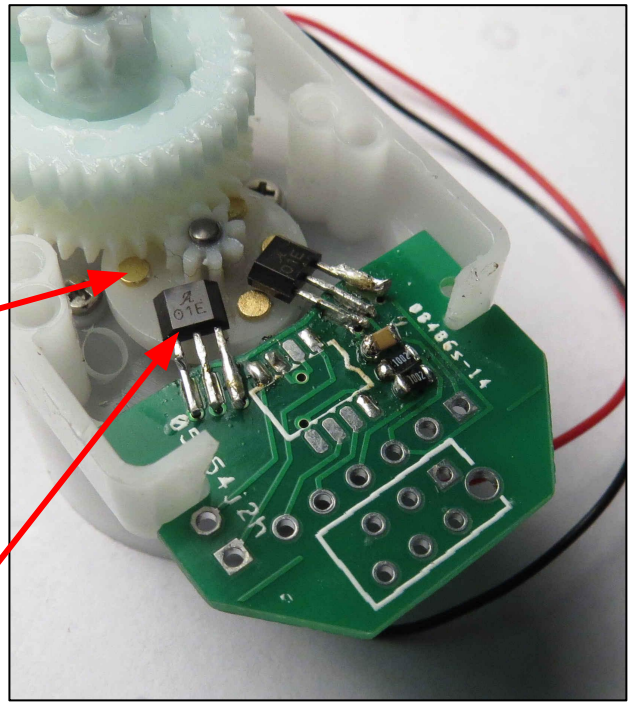




Encodeur magnétique

- Disponibles au local
- Le guide de montage est sur le site
- Seulement compatible avec les moteurs BO-10
- Deux rapports de réduction disponibles





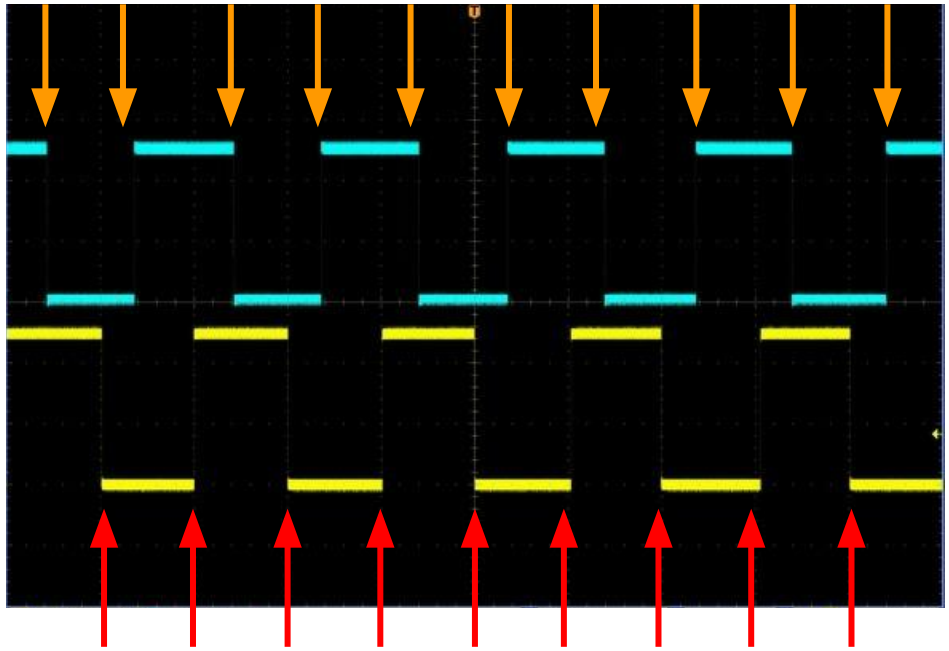
Côté software

- Il faut:
  - **détecter les flancs** montants ou descendants
  - **incrémenter ou décrémenter un compteur** interne en fonction du sens de rotation déterminé

- Deux manières de faire ceci:
  - avec des **interruptions**
  - avec un **timer**

Côté software - interruptions

- Une interruption par capteur
- On configure les interruptions pour détecter les flancs montants et descendants



Signal Capteur 1

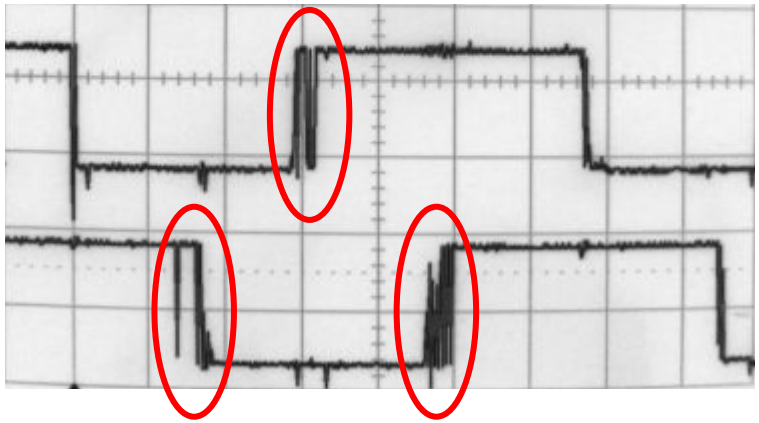
Signal Capteur 2

Déclenchement de l'interruption 1

Déclenchement de l'interruption 2

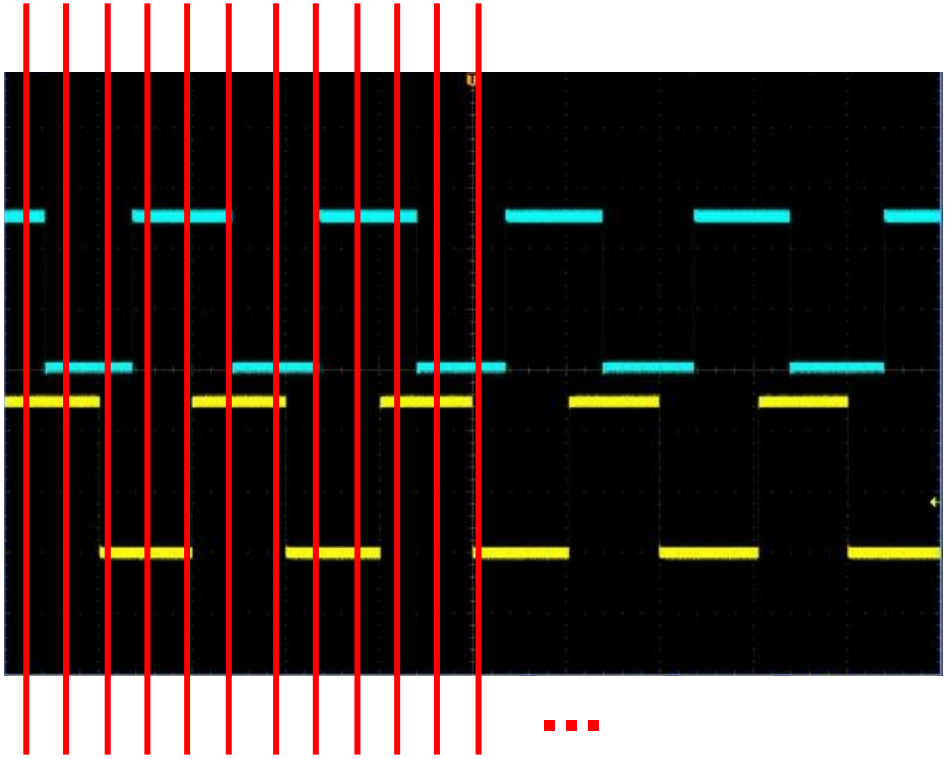
Côté software - interruptions

- Avantages:
  - on ne risque pas de rater un flanc
  - la boucle n'est pas interrompue quand il n'y a pas de flanc (à l'arrêt par ex.)
- Inconvénients:
  - nécessite des **pins spécifiques**
  - très sensible aux **rebonds de contact**



Côté software - timers

- Un lit de manière périodique les valeurs des capteurs pour voir si l'une d'elles a changé



Côté software - timers

- Avantages:
  - **un seul timer** pour un très grand nombre de capteurs, sur **n'importe quels pins**
  - insensible aux rebonds des capteurs
- Inconvénients:
  - interruption inutile du programme principale quand les roues ne bougent pas
  - au delà d'une certaine vitesse, on ne lit plus assez rapidement les valeurs de capteurs

Library Robopoly

- Pour faciliter les choses:
  - une library fait déjà tout: RomEnco
- Implémente la méthode du Timer pour contrôler **jusqu'à 4 encodeurs simultanément**
- Téléchargeable depuis le Git de Robopoly:

github.com/Robopoly/Robopoly_RomeEnco

- Pas encore totalement documentée
- Modifier le Timer à utiliser en décommentant une ligne dans RomEnco.h
  (Timers recommandés: 1 ou 3)

Library Robopoly - fonctions

- `RomEnco encoder()`
  - déclare un objet de type `RomEnco`
  - un objet par encodeur (paire de capteurs)
  - à déclarer en variable globale

- `encoder.begin(pinA, pinB)`
  - commence la lecture avec des capteurs connectés aux pins `pinA` et `pinB`
  - pour le `setup()`

Library Robopoly - fonctions

- `encoder.getPosition()`
  - retourne le compteur d'incréments

- `encoder.resetPosition()`
  - pour remettre à zéro le compteur de pas

- `encoder.getIncrement()`
  - combinaison de `getPosition()` et `resetPosition()`
  - retourne le compteur et le remet à zéro

Library Robopoly - exemple

```
#include <prismino.h>
#include <RomEnco.h>

RomEnco enco;

void setup() {
  setSpeed(50, 50);
  enco.begin(4, 5);
  Serial.begin(9600);
}

void loop() {
  Serial.println(enco.getPosition());
  delay(200);
}
```

Inclure la library

Déclarer un objet encodeur

Commencer la lecture sur les pins 4 et 5

Afficher la valeur du compteur dans le terminal Serial

Quelques applications

- Odométrie
  - se localiser sur un terrain (GC!)
  - atteindre un emplacement précis
  - savoir revenir à son point de départ
  - Lecture: ici et ici
- Estimation de la vitesse du robot
  - Créer des rampes d'accélération et de décélération pour éviter les mouvement trop brusques
  - Détecter des pentes ou des collisions

Mise en garde concernant l'odométrie

- Attention: l'utilisation d'encodeurs est loin d'être la solution idéale!

- Principaux inconvénients:
  - **très sensible au dérapage** des roues (freinage d'urgence, collisions…)
  - si placé avant la réduction du moteur: **ne prend pas en compte le jeu** de celle-ci → grosse incertitude
  - Essayer de se baser sur différents capteurs

Calendrier du semestre

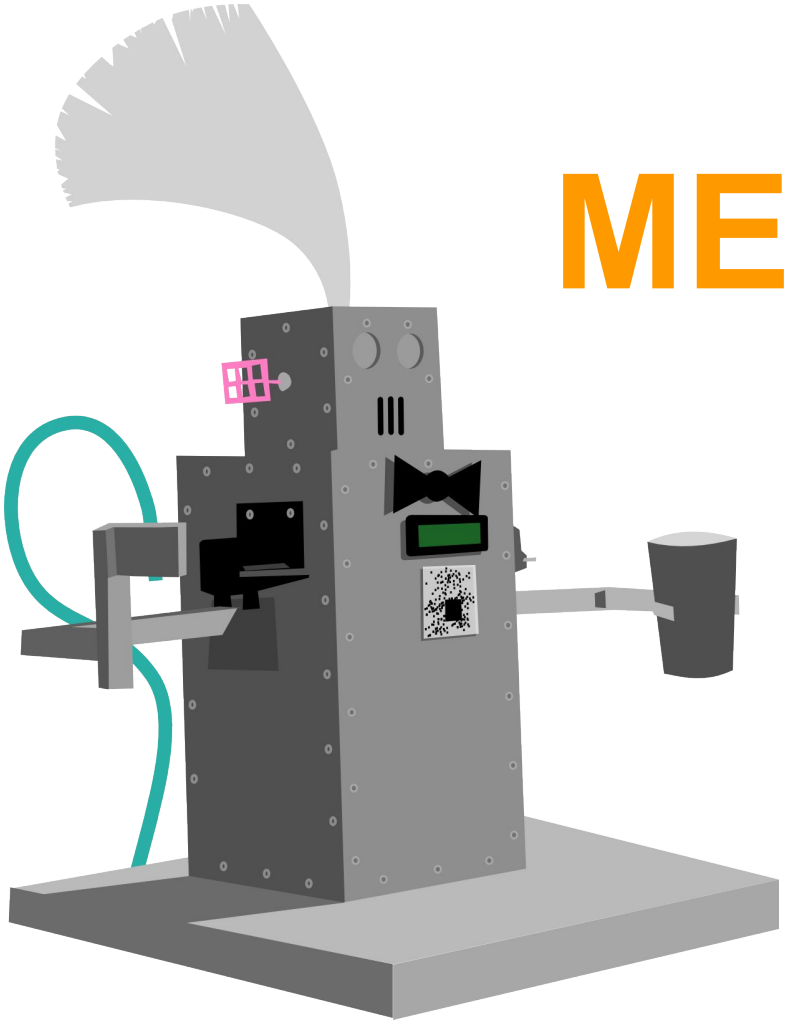
- **Grand Concours**
  - Samedi 19 mars en SG1

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?