

roboonly

The background features a collage of elements related to robotics and electronics: a printed wiring board (PCB) with various components like resistors (R32, R33, R34, R37, R38, R39, R40, R41, R42, R43, R44, R45, R46, R47, R48, R49, R50, R51, R52, R53, R54, R55, R56, R57, R58, R59, R60, R61, R62, R63, R64, R65, R66, R67, R68, R69, R70, R71, R72, R73, R74, R75, R76, R77, R78, R79, R80, R81, R82, R83, R84, R85, R86, R87, R88, R89, R90, R91, R92, R93, R94, R95, R96, R97, R98, R99, R100), capacitors (C1, C2, C3, C4, C5, C6, C7, C8, C9, C10, C11, C12, C13, C14, C15, C16, C17, C18, C19, C20, C21, C22, C23, C24, C25, C26, C27, C28, C29, C30, C31, C32, C33, C34, C35, C36, C37, C38, C39, C40, C41, C42, C43, C44, C45, C46, C47, C48, C49, C50, C51, C52, C53, C54, C55, C56, C57, C58, C59, C60, C61, C62, C63, C64, C65, C66, C67, C68, C69, C70, C71, C72, C73, C74, C75, C76, C77, C78, C79, C80, C81, C82, C83, C84, C85, C86, C87, C88, C89, C90, C91, C92, C93, C94, C95, C96, C97, C98, C99, C100), and integrated circuits (U1, U2, U3, U4, U5, U6, U7, U8, U9, U10, U11, U12, U13, U14, U15, U16, U17, U18, U19, U20, U21, U22, U23, U24, U25, U26, U27, U28, U29, U30, U31, U32, U33, U34, U35, U36, U37, U38, U39, U40, U41, U42, U43, U44, U45, U46, U47, U48, U49, U50, U51, U52, U53, U54, U55, U56, U57, U58, U59, U60, U61, U62, U63, U64, U65, U66, U67, U68, U69, U70, U71, U72, U73, U74, U75, U76, U77, U78, U79, U80, U81, U82, U83, U84, U85, U86, U87, U88, U89, U90, U91, U92, U93, U94, U95, U96, U97, U98, U99, U100). There is also a snippet of C++ code: 

```
begin(9600);  
LED, OUTPUT);  
  
int potentiometer and button values  
int potentiometer;  
int button;  
void setup()  
{  
  Serial.begin(9600);  
  pinMode(10, INPUT);  
  pinMode(11, INPUT);  
  pinMode(12, INPUT);  
  pinMode(13, INPUT);  
  pinMode(14, INPUT);  
  pinMode(15, INPUT);  
  pinMode(16, INPUT);  
  pinMode(17, INPUT);  
  pinMode(18, INPUT);  
  pinMode(19, INPUT);  
  pinMode(20, INPUT);  
  pinMode(21, INPUT);  
  pinMode(22, INPUT);  
  pinMode(23, INPUT);  
  pinMode(24, INPUT);  
  pinMode(25, INPUT);  
  pinMode(26, INPUT);  
  pinMode(27, INPUT);  
  pinMode(28, INPUT);  
  pinMode(29, INPUT);  
  pinMode(30, INPUT);  
  pinMode(31, INPUT);  
  pinMode(32, INPUT);  
  pinMode(33, INPUT);  
  pinMode(34, INPUT);  
  pinMode(35, INPUT);  
  pinMode(36, INPUT);  
  pinMode(37, INPUT);  
  pinMode(38, INPUT);  
  pinMode(39, INPUT);  
  pinMode(40, INPUT);  
  pinMode(41, INPUT);  
  pinMode(42, INPUT);  
  pinMode(43, INPUT);  
  pinMode(44, INPUT);  
  pinMode(45, INPUT);  
  pinMode(46, INPUT);  
  pinMode(47, INPUT);  
  pinMode(48, INPUT);  
  pinMode(49, INPUT);  
  pinMode(50, INPUT);  
  pinMode(51, INPUT);  
  pinMode(52, INPUT);  
  pinMode(53, INPUT);  
  pinMode(54, INPUT);  
  pinMode(55, INPUT);  
  pinMode(56, INPUT);  
  pinMode(57, INPUT);  
  pinMode(58, INPUT);  
  pinMode(59, INPUT);  
  pinMode(60, INPUT);  
  pinMode(61, INPUT);  
  pinMode(62, INPUT);  
  pinMode(63, INPUT);  
  pinMode(64, INPUT);  
  pinMode(65, INPUT);  
  pinMode(66, INPUT);  
  pinMode(67, INPUT);  
  pinMode(68, INPUT);  
  pinMode(69, INPUT);  
  pinMode(70, INPUT);  
  pinMode(71, INPUT);  
  pinMode(72, INPUT);  
  pinMode(73, INPUT);  
  pinMode(74, INPUT);  
  pinMode(75, INPUT);  
  pinMode(76, INPUT);  
  pinMode(77, INPUT);  
  pinMode(78, INPUT);  
  pinMode(79, INPUT);  
  pinMode(80, INPUT);  
  pinMode(81, INPUT);  
  pinMode(82, INPUT);  
  pinMode(83, INPUT);  
  pinMode(84, INPUT);  
  pinMode(85, INPUT);  
  pinMode(86, INPUT);  
  pinMode(87, INPUT);  
  pinMode(88, INPUT);  
  pinMode(89, INPUT);  
  pinMode(90, INPUT);  
  pinMode(91, INPUT);  
  pinMode(92, INPUT);  
  pinMode(93, INPUT);  
  pinMode(94, INPUT);  
  pinMode(95, INPUT);  
  pinMode(96, INPUT);  
  pinMode(97, INPUT);  
  pinMode(98, INPUT);  
  pinMode(99, INPUT);  
  pinMode(100, INPUT);  
}
```

Bienvenue !

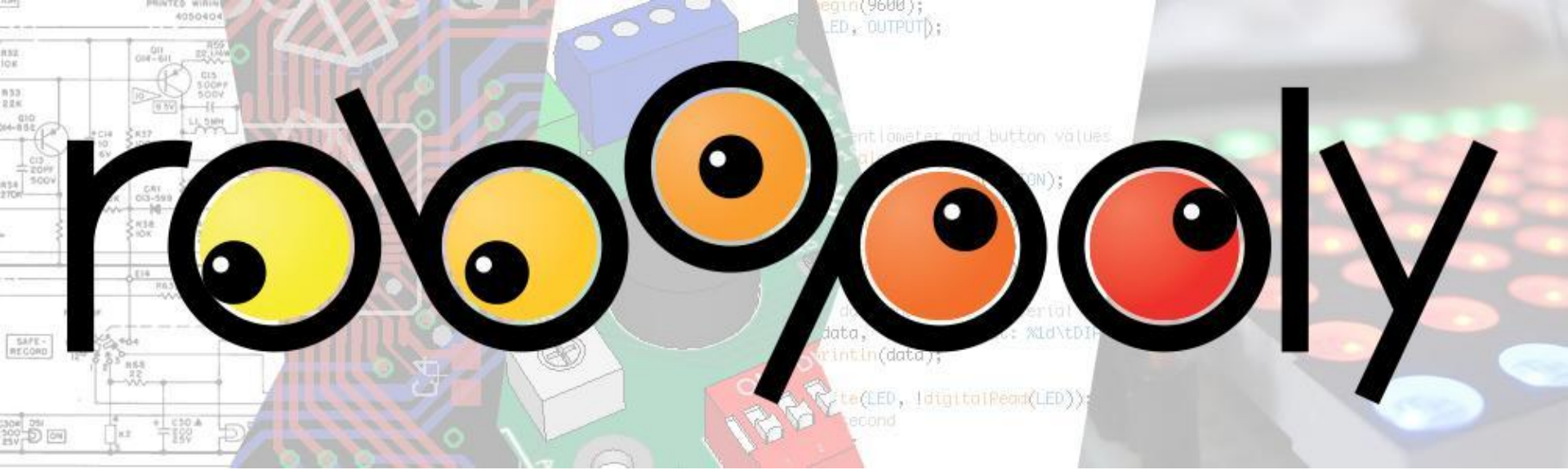
# Programme du jour

## ❖ **Partie 1 : Bases d'électronique**

- Quelques définitions
- Principaux composants
- Soudure

## ❖ **Partie 2 : Microcontrôleur**

- Qu'est-ce donc ?
- Prismo
- Comment programmer le uC ?



roboonly

# Bases d'électronique

## Microcontrôleur

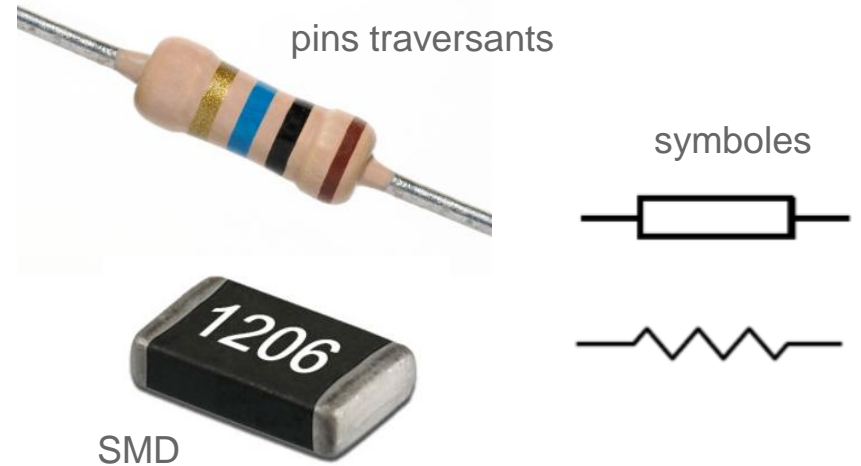
# Courant et tension électrique

- On utilise l'analogie avec un fleuve\*
- **Courant**
  - correspond au nombre de charges électriques (électrons) par seconde
  - assimilé au débit du fleuve
  - Unité : ampère (A)
- **Tension**
  - différence de potentiel entre deux points
  - assimilée au dénivelé (nécessaire au déplacement du fluide)
  - utilisée pour transmettre un signal
  - Unité : volt (V)

# Principaux composants

- **Résistance**

- dissipe l'énergie électrique sous forme de chaleur
- utilisée pour **limiter le courant**



- **LED** (light-emitting diode)
  - produit de la **lumière**
  - souvent associée à une résistance



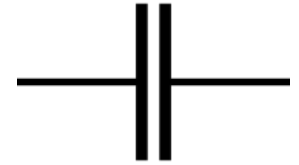
*SMD = surface mounted device*

# Principaux composants

- **Capacité**
  - agit comme un **réservoir** de charges électriques



pins traversants



SMD

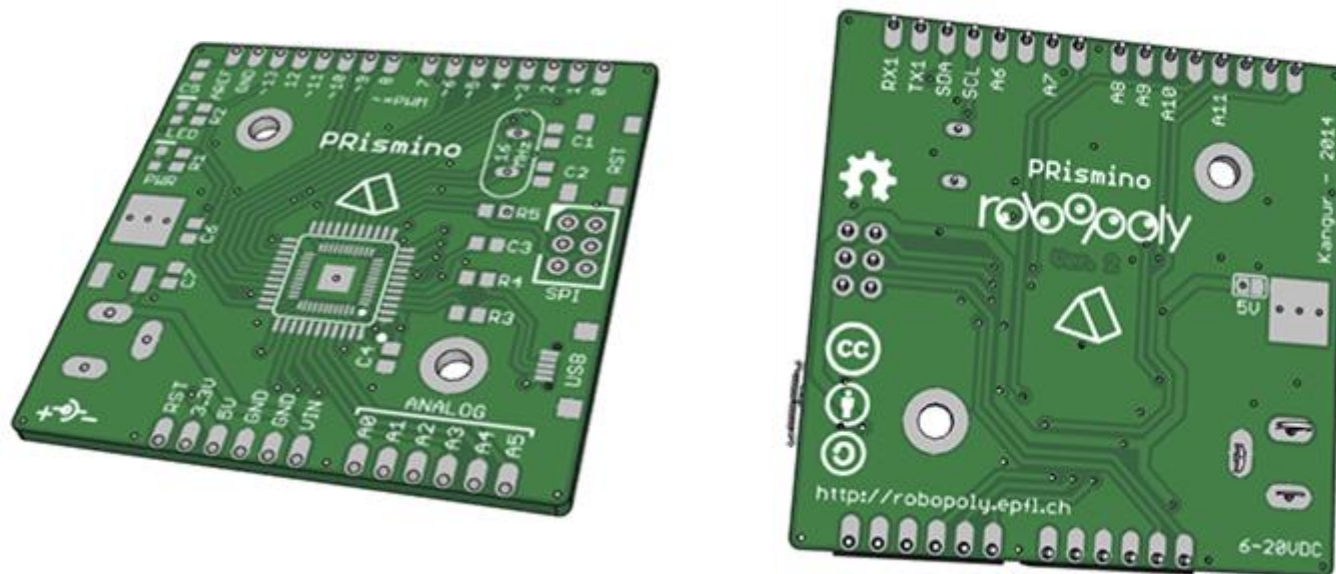
- **Transistor**
  - à la base de l'électronique d'aujourd'hui
  - intéressant mais très compliqué



pins traversants

# Le PCB

- Pour Printed Circuit Board = **circuit imprimé**
- C'est sur celui-ci que sont soudés et reliés électriquement les composants électroniques



## Soudure

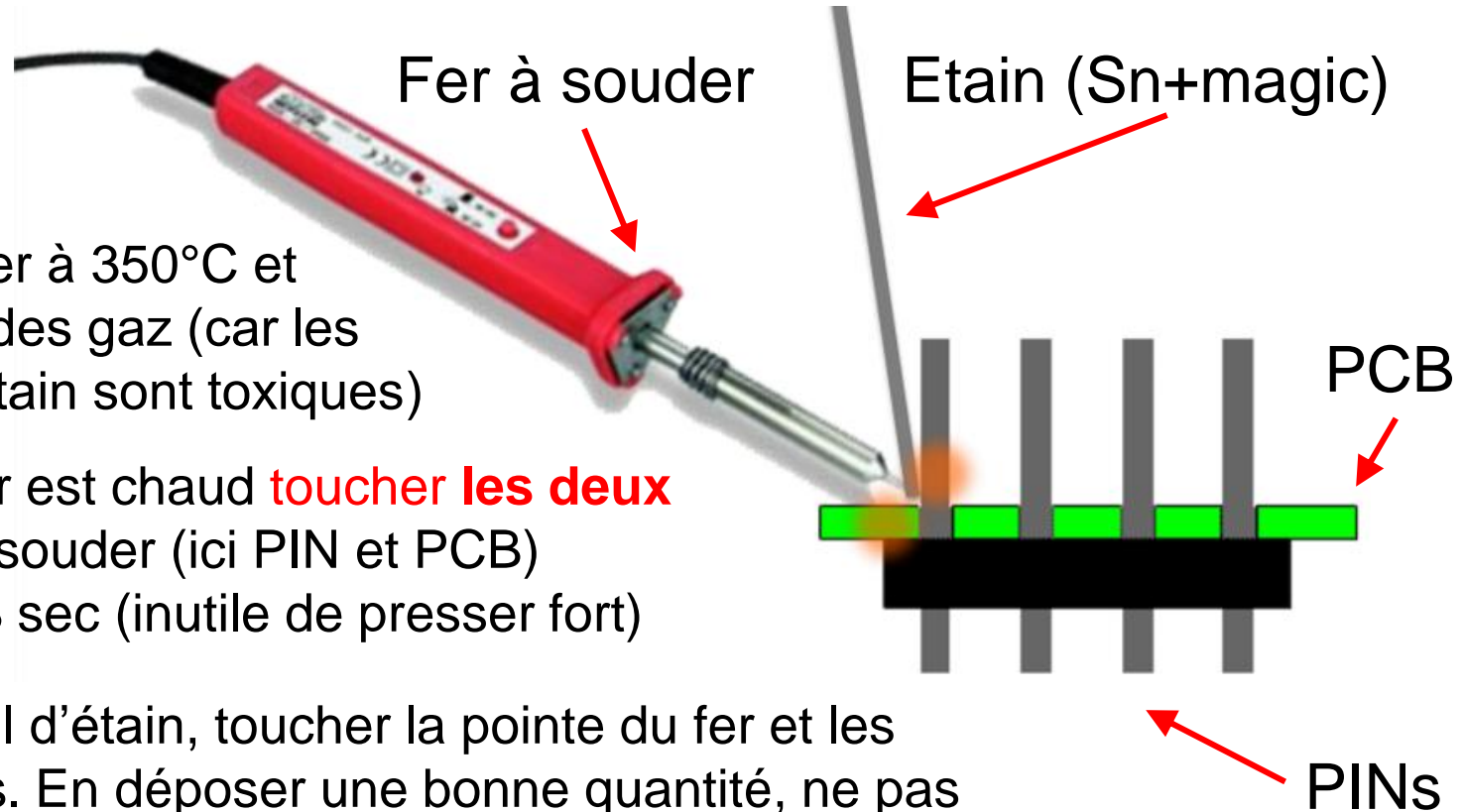


L'étain contient de **nombreuses substances toxiques** qui se dégagent dans l'air quand il fond.

Il est donc **obligatoire** d'**utiliser la ventilation** pour aspirer la fumée !



# Soudure



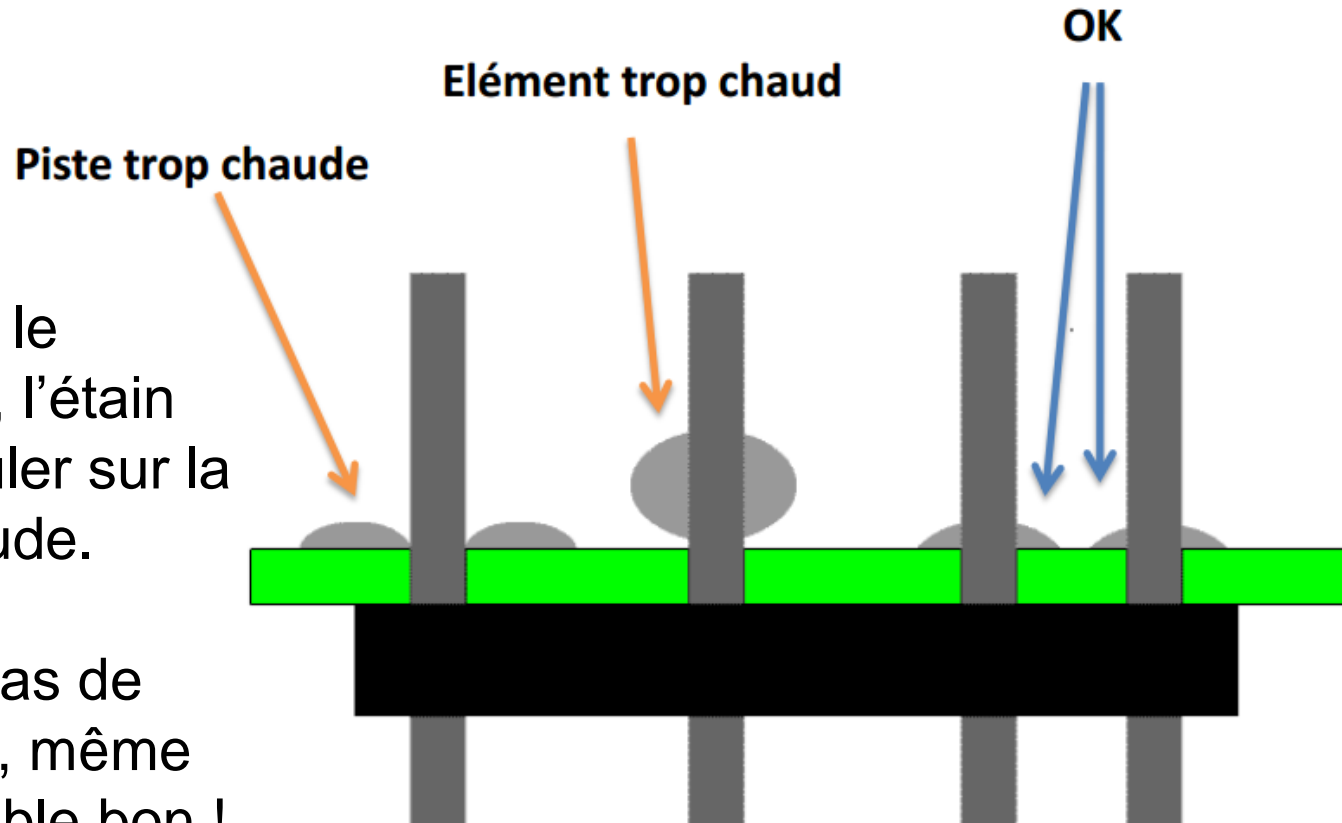
- Allumer le fer à 350°C et l'aspiration des gaz (car les vapeurs d'étain sont toxiques)
- Quand le fer est chaud **toucher les deux éléments** à souder (ici PIN et PCB) pendant 2-3 sec (inutile de presser fort)
- Amener le fil d'étain, toucher la pointe du fer et les composants. En déposer une bonne quantité, ne pas être radin ! Garder le fer **maximum 10-15 secondes** sur les composants
- Enlever le fer, juger la qualité de la soudure (éventuellement retoucher)  
Ne pas poser le doigt dessus !

# Une belle soudure

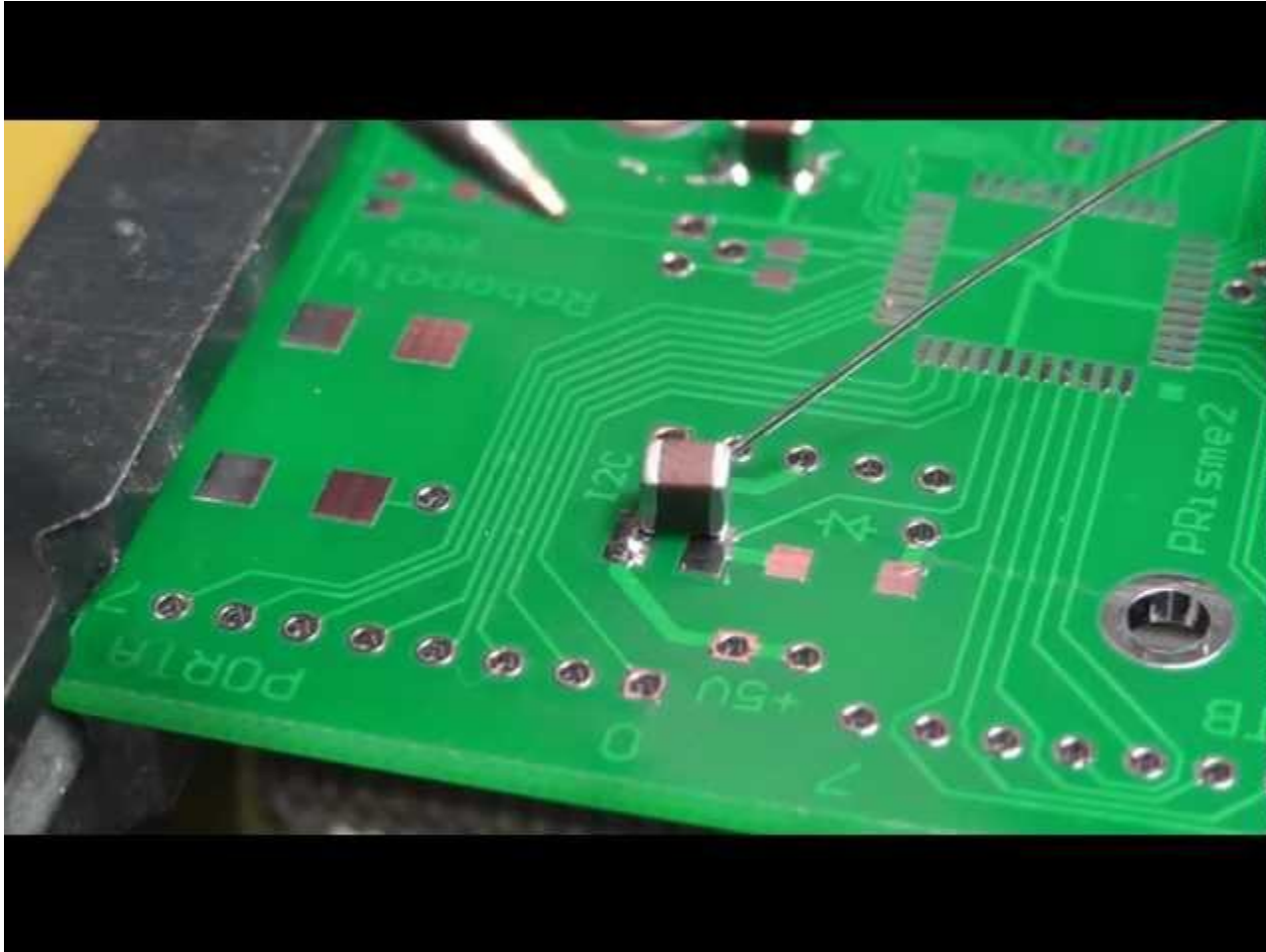
Il faut que le fer à souder touche la partie à souder (ici un PIN) **ET** le support (ici le PCB)

Si on chauffe trop le support ou le PIN, l'étain va venir s'accumuler sur la partie la plus chaude.

Souvent, il n'y a pas de contact électrique, même si à l'oeil tout semble bon !

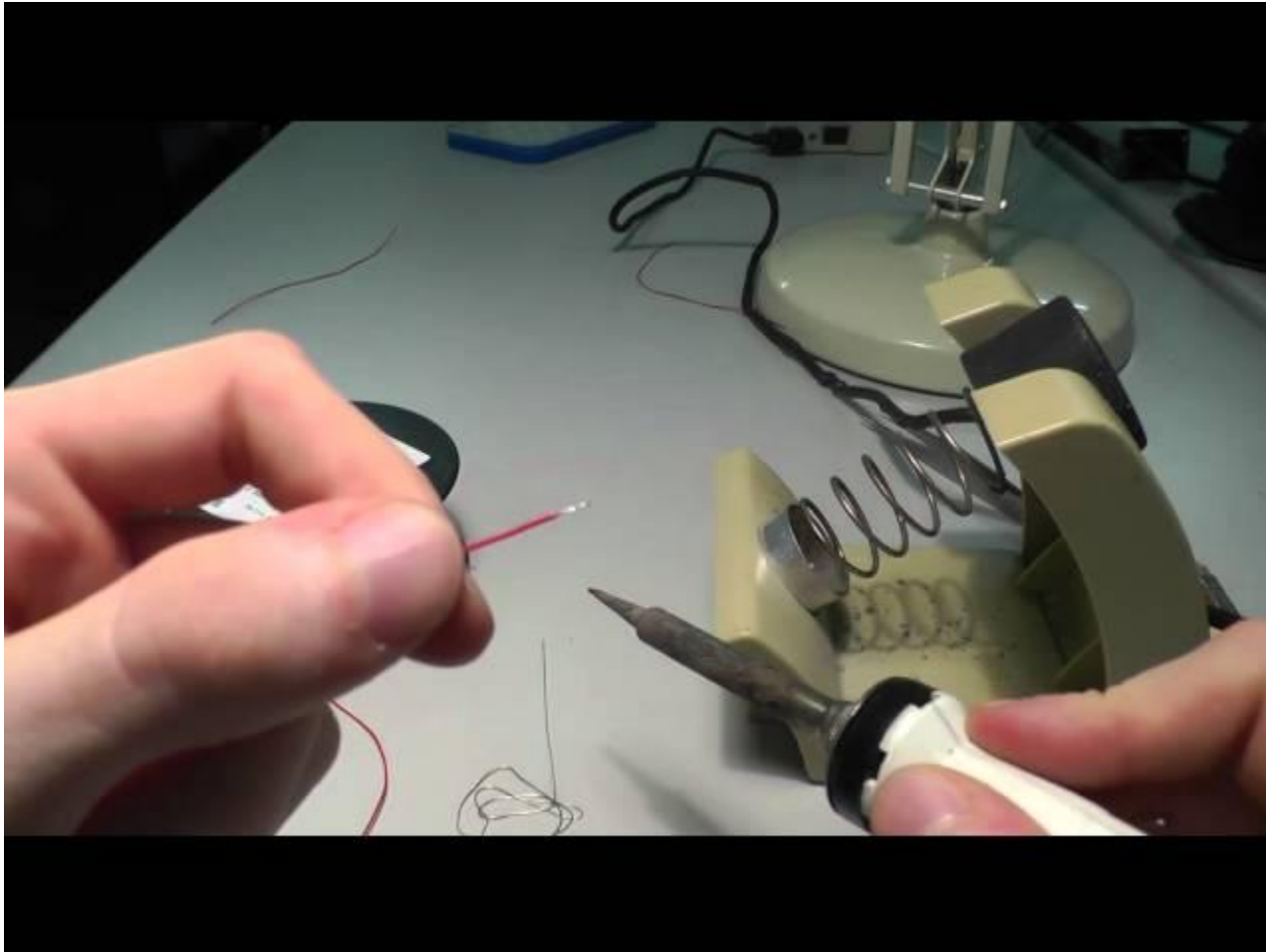


# Soudure d'un SMD



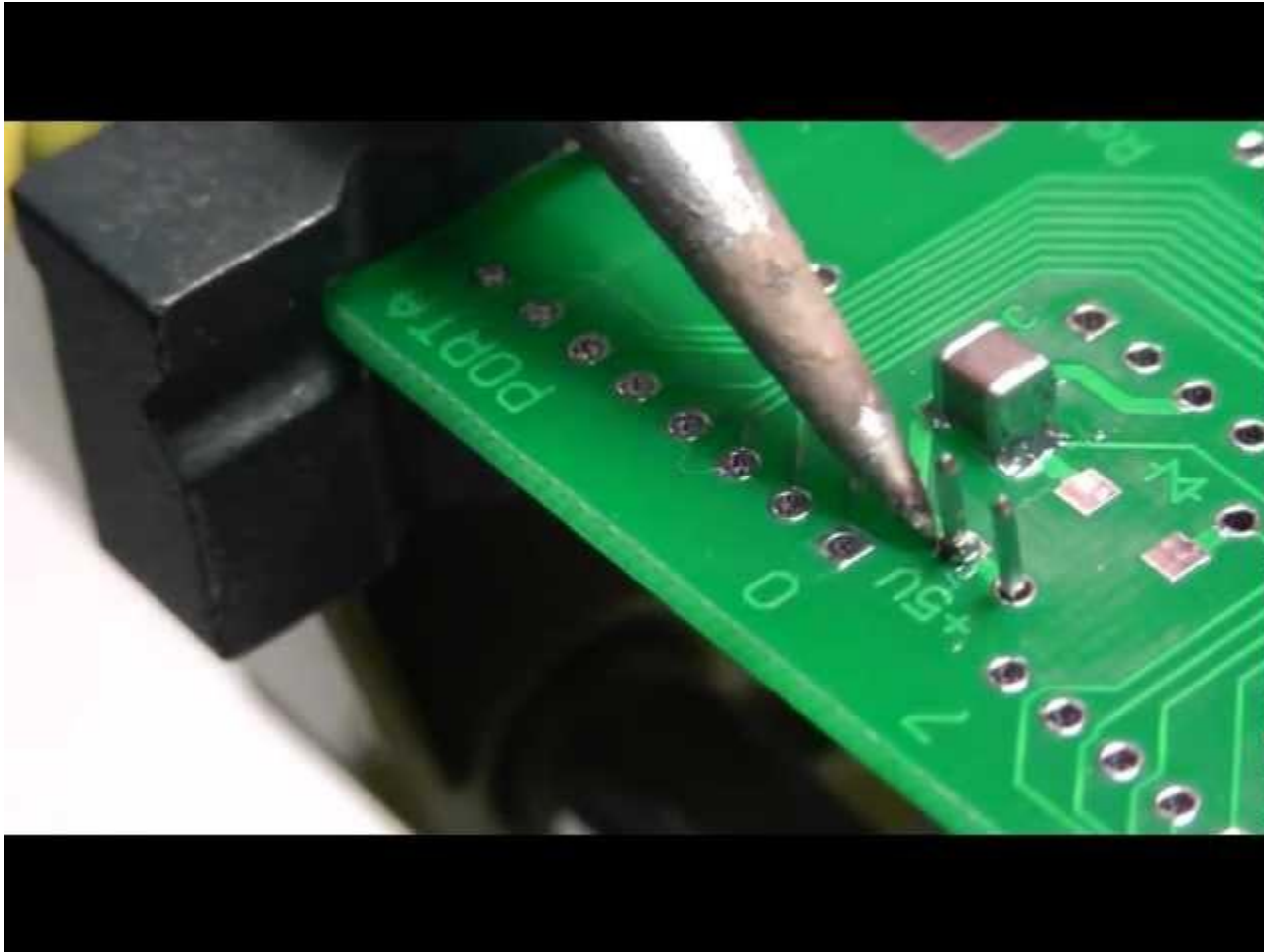
<https://youtu.be/Ewz7TdvhHxo>

# Etamage et soudure d'un fil



[https://youtu.be/d1\\_FzzFbH9I](https://youtu.be/d1_FzzFbH9I)

# Soudure d'un PIN



<https://youtu.be/gbA89JW9-oA>

Une dernière qui résume tout [ici](#)

# Soudure des modules

**Pas besoin de comprendre  
comment marche chaque  
module pour commencer à  
souder son robot !**

Les instructions de montage sont disponibles ici :  
[robopoly.epfl.ch/prisme](https://robopoly.epfl.ch/prisme)



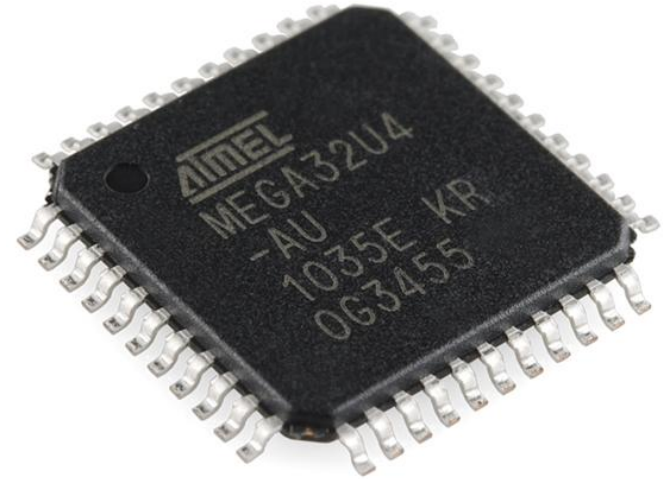
robotology

# Bases d'électronique

## Microcontrôleur

# Qu'est-ce qu'un microcontrôleur ?

- Abrégé uC ou MCU (MicroController Unit)
- Puce électronique digitale



- Réunit dans un seul boîtier les fonctions suivantes :
  - unité centrale de traitement **CPU** (Central Process Unit)
  - circuits d'**entrée/sortie** (périphériques)
  - **mémoire** (programme et données)
- Bref comme un ordinateur mais en moins rapide



# PRismino

## Microcôntroleur ATmega32u4

## Bouton RESET

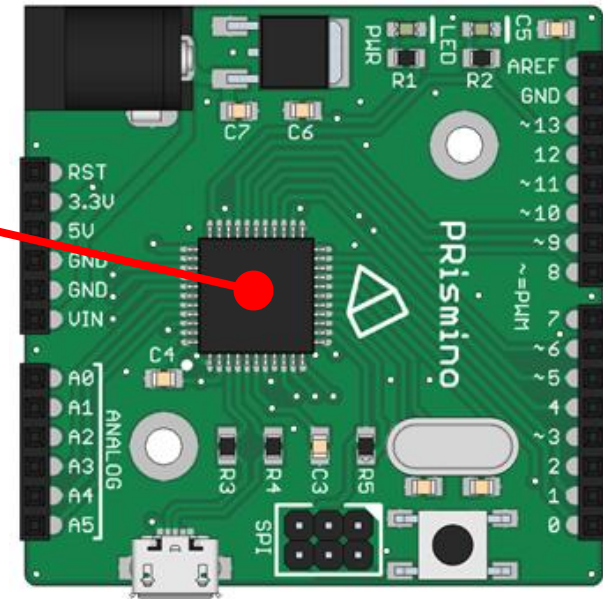
- permet de réinitialiser le programme

## GPIO (General Purpose Input Output)

- sorties/entrées digitales ou analogiques (de 0V à 5V)
  - en sortie : contrôle de périphériques, actions complexes
  - en entrée : lecture de valeurs (interrupteur, capteur)

## Micro-USB

- interface avec le “vrai” ordinateur
- permet de charger les programmes



# PRismino

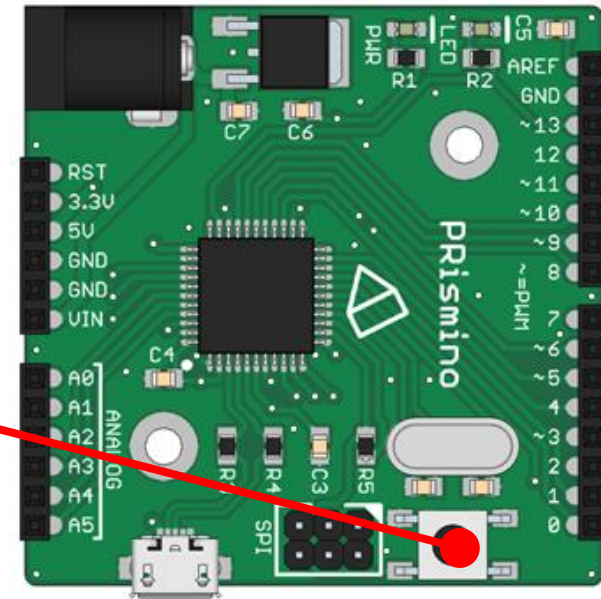
Microcôntroleur ATmega32u4

## Bouton RESET

- permet de réinitialiser le programme

## GPIO (General Purpose Input Output)

- sorties/entrées digitales ou analogiques (de 0V à 5V)
  - en sortie : contrôle de périphériques, actions complexes
  - en entrée : lecture de valeurs (interrupteur, capteur)



## Micro-USB

- interface avec le “vrai” ordinateur
- permet de charger les programmes

# PRismino

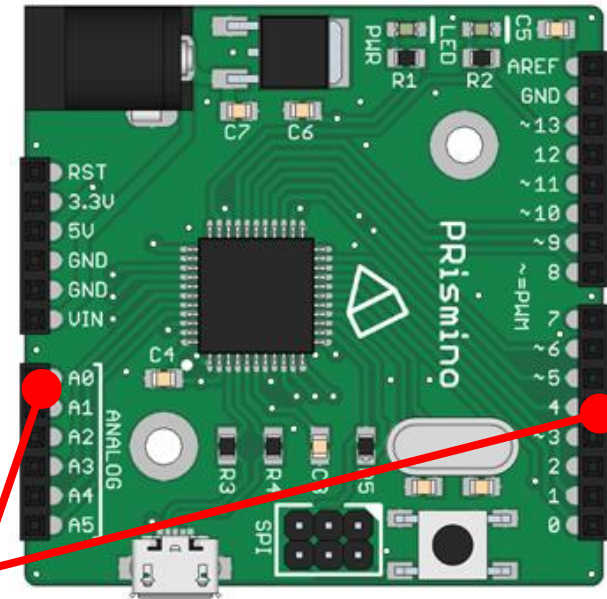
Microcôntroleur ATmega32u4

Bouton RESET

- permet de réinitialiser le programme

**GPIO (General Purpose Input Output)**

- sorties/entrées digitales ou analogiques (de 0V à 5V)
  - en sortie : contrôle de périphériques, actions complexes
  - en entrée : lecture de valeurs (interrupteur, capteur)



Micro-USB

- interface avec le “vrai” ordinateur
- permet de charger les programmes

# PRismino

Microcôntroleur ATmega32u4

Bouton RESET

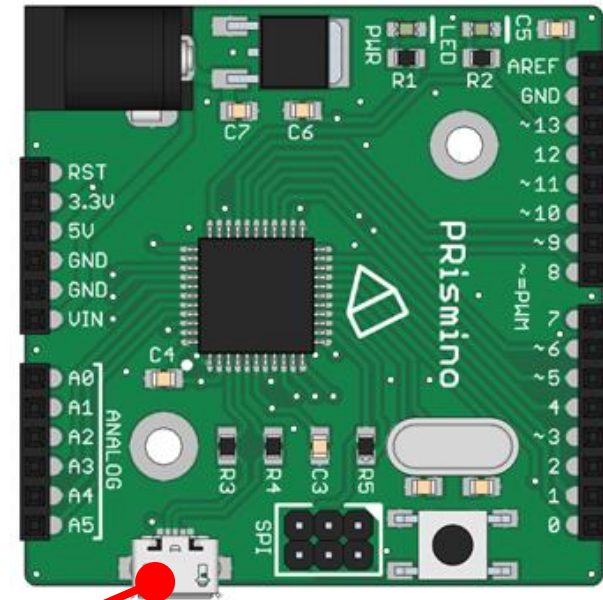
- permet de réinitialiser le programme

GPIO (General Purpose Input Output)

- sorties/entrées digitales ou analogiques (de 0V à 5V)
  - en sortie : contrôle de périphériques, actions complexes
  - en entrée : lecture de valeurs (interrupteur, capteur)

**Micro-USB**

- interface avec le “vrai” ordinateur
- permet de charger les programmes



# Bootloader

- Petit programme qui permet de programmer le microcontrôleur par USB
- A faire une fois que tout est soudé
- Utiliser le petit boîtier présent au local



Comment l'utiliser ? Vidéo : <https://www.youtube.com/watch?v=6KTRMYtDIAE>

# Comment programmer le uC ?

- On utilise l'environnement Arduino
- Programmation en C, C++, Assembleur
- Composé d'un éditeur de code, un compilateur et une bibliothèque de fonctions de base

→ A télécharger ici :

<https://www.arduino.cc/en/Main/Software>

→ Documentation :

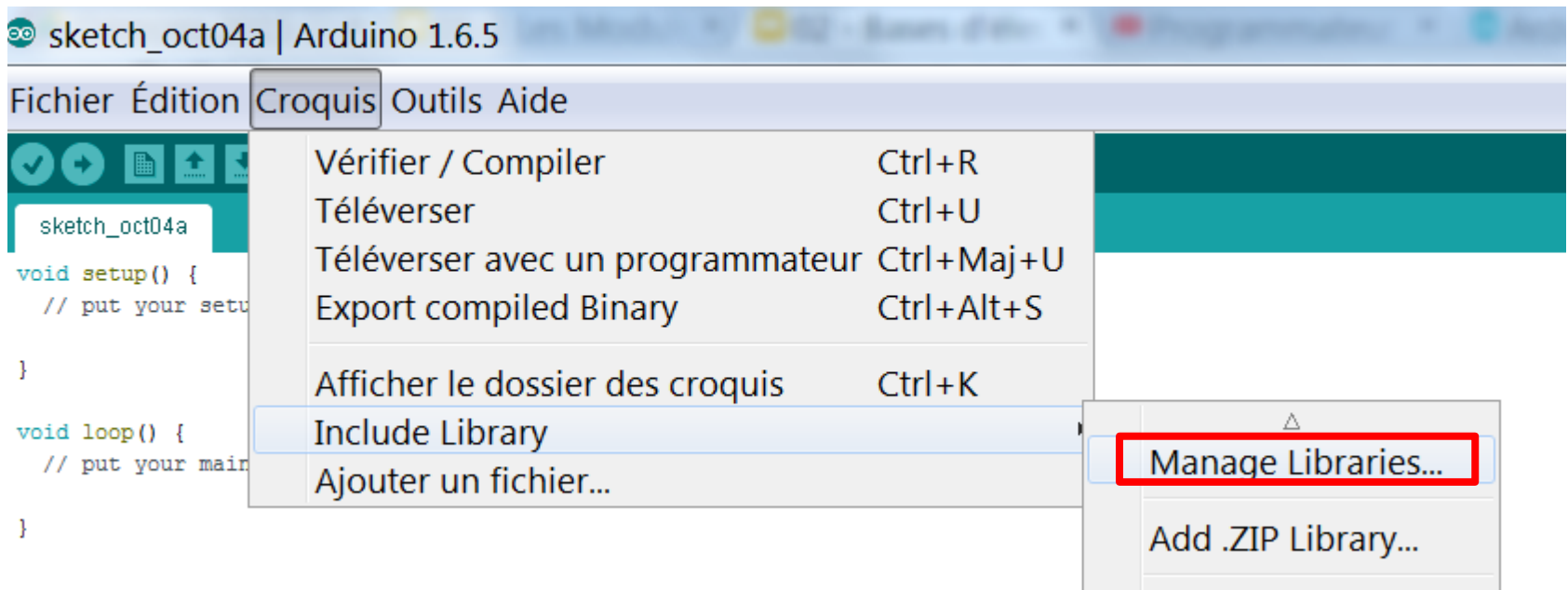
[www.arduino.cc/en/Reference/HomePage](http://www.arduino.cc/en/Reference/HomePage)

→ Bibliothèque de Robopoly :

[github.com/Robopoly/Robopoly\\_PRismino](https://github.com/Robopoly/Robopoly_PRismino)

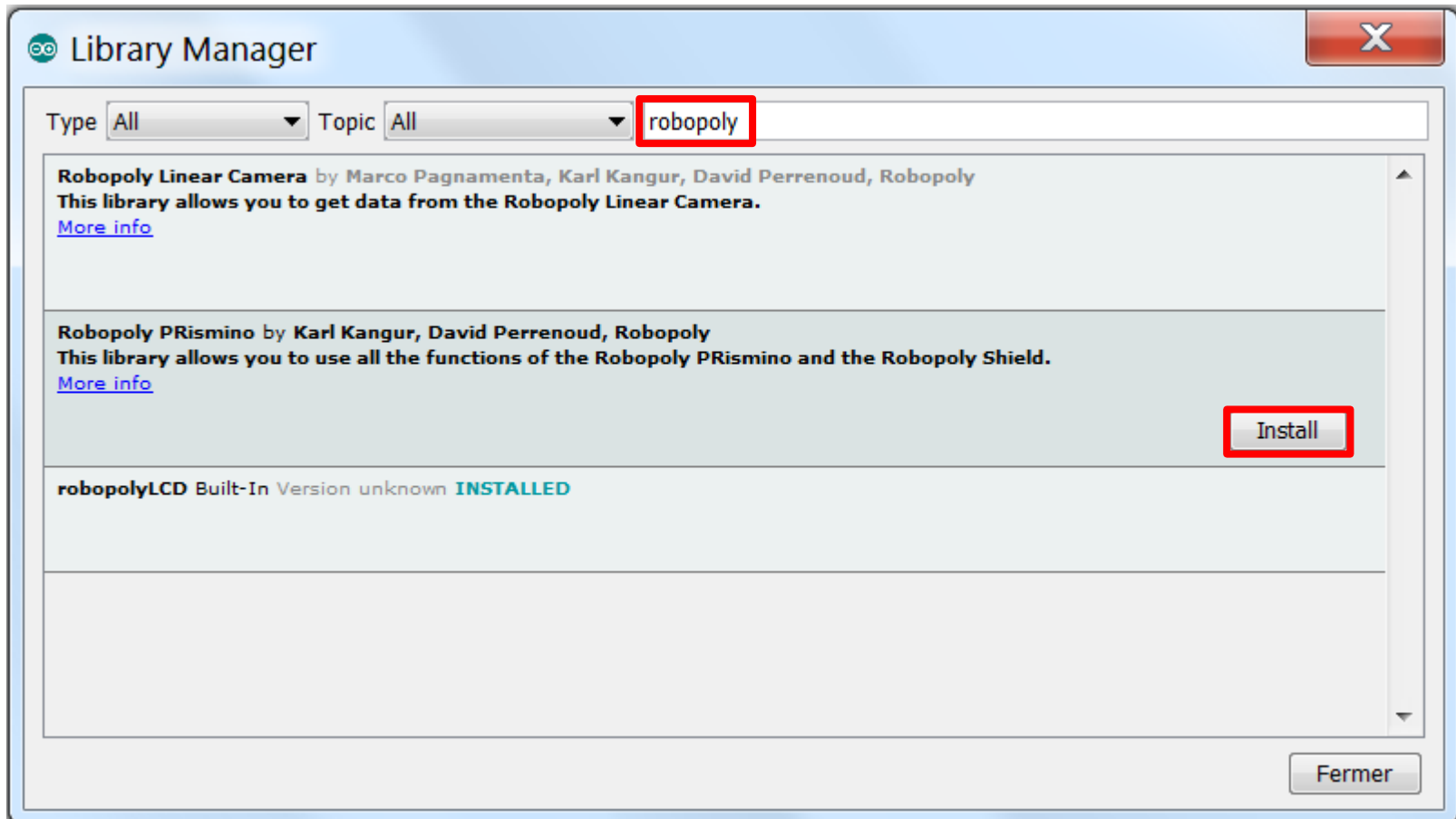
# Installer la bibliothèque Robopoly

Automatiquement depuis le logiciel Arduino



# Installer la bibliothèque Robopoly

Automatiquement depuis le logiciel Arduino





# Prochains événements

- **Workshop number ONE !**
  - Samedi 14 octobre de 9h à 18h
  - Au BM 5 202
  - Pour commencer à monter ton robot, discuter de tes projets, demander de l'aide au comité !
  - Inscription doodle (=> lien aussi envoyé par email)  
<https://doodle.com/poll/t9w9ysp5a8iddnma>
  
- **Prochaine démon**
  - Lundi prochain, 12h15, ELA1
  - Début de programmation, module Shield



## Contact / Infos

**Contact principal**

[robopoly@epfl.ch](mailto:robopoly@epfl.ch)

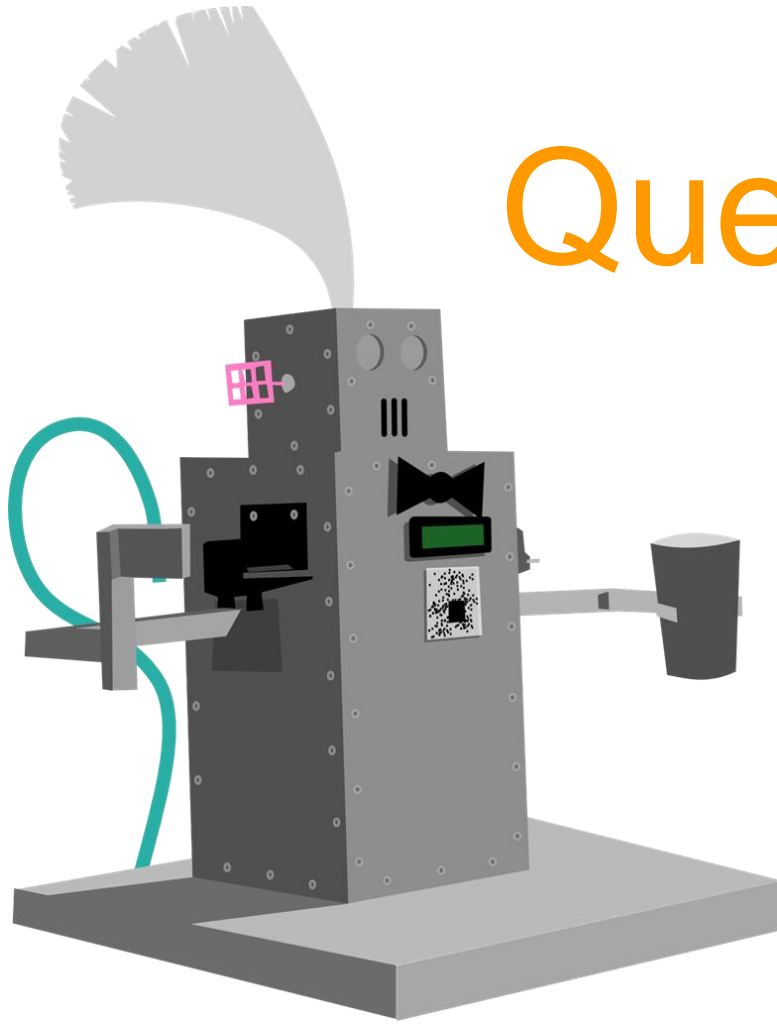
**Site officiel - toutes les infos et slides sont là !**

[robopoly.epfl.ch](http://robopoly.epfl.ch)

**Facebook - pour suivre l'actualité du club !**

[www.facebook.com/robopoly](http://www.facebook.com/robopoly)

# Questions ?



# MERCI !