



roboonly

The background features a collage of images related to robotics and electronics. On the left, there is a printed wiring board (PCB) with various components labeled, including resistors (R32, R33, R34, 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 (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). In the center, a blue servo motor is visible. On the right, there is a close-up of a keyboard with several keys highlighted in blue and red. Overlaid on the background are snippets of code, including `begin(9600);`, `LED, OUTPUT);`, `intrometer and button values`, `ON);`, `data,`, `: %ld\\tD\\t`, `println(data);`, `te(LED, !digitalRead(LED));`, and `second`.

SERVOMOTEURS

PROGRAMMES TYPES

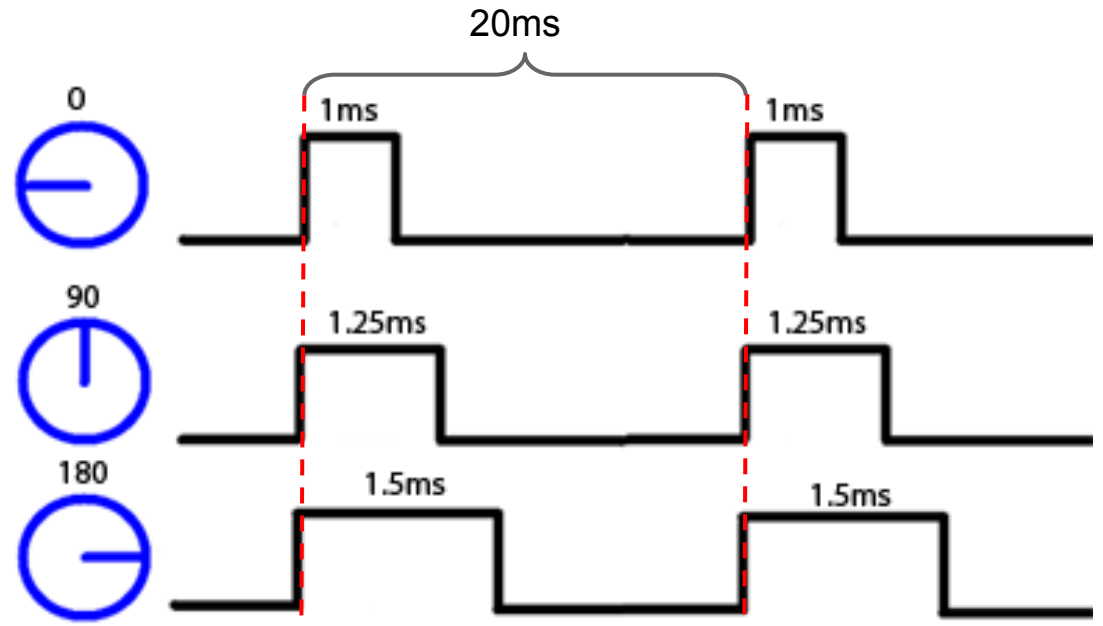
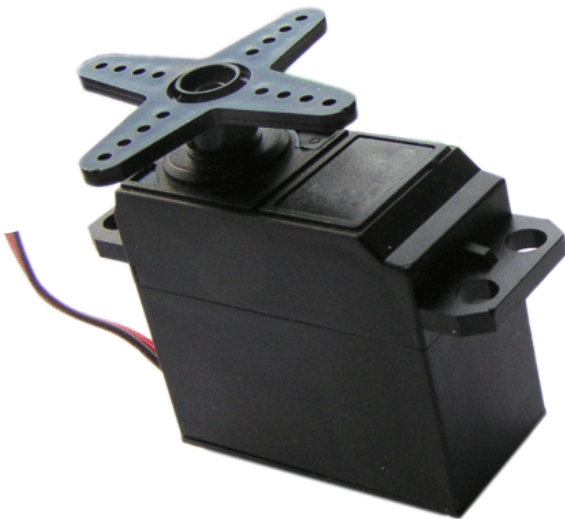
Servomoteurs

- Moteur contrôlé en position (angle)
 - entre 0° et 180°
 - sur l'entièreté du tour

- Servomoteur du kit
 - contrôle par PWM
 - régulation de position interne
 - consigne envoyée par PWM

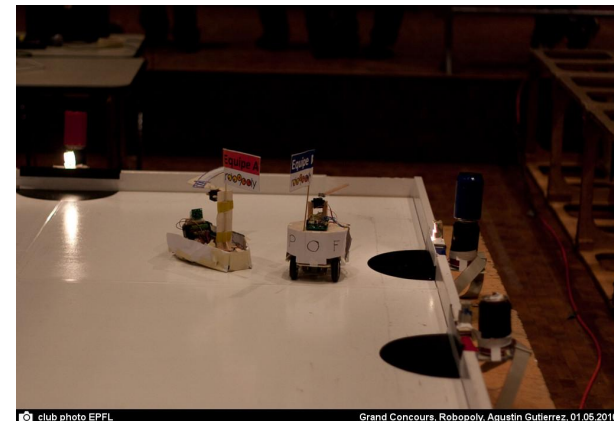
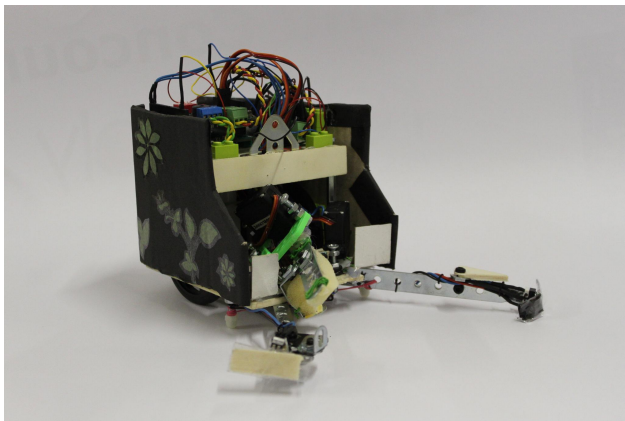
Servomoteur - principe de fonctionnement

Exemple d'utilisation: Servomoteur

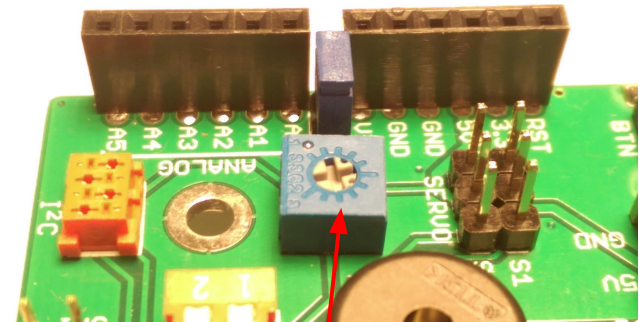


Servomoteur - à quoi ça sert

- Indication de niveau
- Prehension (attrapper des objets)
- Chutter des objets



Servomoteur - démonstration



Préparer le servomoteur
(dire que ça existe)

```
#include <Servo.h>

// define the servo motor instances here
Servo my_servo;
uint16_t ir_value;
uint8_t servo_pos;
```

Dire que le servomoteur
est sur le pin 6 (S1 sur le
silkscreen)

```
void setup()
{
  my_servo.attach(6);
}

void loop()
{
  ir_value = analogRead(A0);

  servo_pos = map(ir_value, 0, 1023, 0, 180);

  my_servo.write(servo_pos);
}
```

Lire la valeur analogique
du potentiomètre

Transformer la valeur entre 0
et 1023 en 0 et 180

Appliquer la position sur le
servomoteur

DEMO



roboonly

SERVOMOTEURS

PROGRAMMES TYPES

Programmation - rappels

- for, if/else, while, switch

```

unsigned int i;
for(i = 0; i < 10; i++)
{
}

for(i = 10; i > 0; i--)
{
}

if(condition1)
{
}
else if(condition2)
{
}
else
{
}

while(condition)
{
    if(something happens)
    {
        break;
    }
}

switch(value)
{
    case VALUE1:
        // do something
        break;
    case VALUE1:
        // do something
        break;
    case VALUE1:
        // do something
        break;
    default:
        // do something
}

```


Programmation - Lecture IR

```
// button not pressed
if(btn)
{
    digitalWrite(13,1);           // turn on IR emitter
    delay(1);                     // time for switch on (may be way shorter, like 100us)
    ir_value = analogRead(A2);    // read IR sensor
    digitalWrite(13,0);          // turn off IR emitter (save battery)
    ir_value = 1023 - ir_value;    // to have the same scale as the other example
}
```

Programmation - Lecture IR

```

// button pressed
else
{
    digitalWrite(13,1);           // turn on IR emitter
    delay(1);                     // time for switch on (may be way shorter, like 100us)
    ir_value_on = analogRead(A2); // read IR sensor
    digitalWrite(13,0);          // turn off IR emitter (save battery)
    delay(1);                     // time for switch off (may be way shorter, like 100us)
    ir_value_off = analogRead(A2); // read IR sensor

    ir_value = ir_value_off - ir_value_on; // compute differential
}

```

Programmation - Moteurs

- Utiliser `setSpeed()` + `delay()`

```
#include <prismino.h>

void setup()
{
}

void loop()
{
  setSpeed(20, 20);
}
```

```
#include <prismino.h>

void setup()
{
}

void loop()
{
  setSpeed(20, 20);
  delay(500);
  setSpeed(-20, -20);
  delay(500);
}
```

QUOI FAIRE?

Un robot!

- fini de souder?
- construire la mécanique!

- Défis base de programmation
 - suivi de mur
 - suivi de ligne
 - évitement d'obstacle

- Hésitez pas à utiliser les éléments d'obstacle et les petits mur blancs du local dans l'arène

Arène de test



Infos!

- lundi 17 novembre: **Invité - JD Nicoud**
ancien professeur EPFL
- lundi 24 novembre: **Timers, interruptions,
masquage**