

RAPPORT DE PROJET

ROBOT POLYDOG



Fig. 1 : Le PolyDog dans son habitat naturel

SOMMAIRE

Introduction	3
Planification / Réalisation		
2.1	Fonctionnalité du cahier des charges.....	4
2.2	Fonctionnalité optionnelle	4
2.3	Diagramme de Gantt et Planning réel.....	5
Hardware		
3.1	Pièces 3D	6
3.2	Fonctionnement de la patte.....	7
3.3	Composants électroniques.....	7
Software		
4.1	Classes du Programme	8
4.2	Algorigramme	8
Conclusion	9
Perspective	9
Bibliographie	10

Introduction

Dans le cadre du cours d'Électronique-Arduino de la 2^{ème} année du cycle préparatoire à l'école de Polytech Nice Sophia, nous avons eu l'occasion de créer de A à Z un projet de notre choix. Le temps imparti pour cette réalisation était de 8 séances de TD étalées sur un peu plus de 3 mois. Cependant, nous avons commencé à réfléchir à ce que l'on avait le plus envie de construire un mois avant le début des cours.

Le but de ce projet était de nous donner le goût d'innover grâce à des technologies que nous n'utilisons pas telles que la découpe laser, l'impression 3D ou le fer à souder. Il avait pour vocation de nous former au travail en équipe, ainsi que nous permettre de mettre un pied dans le métier d'ingénieur. Pour ce faire, le projet a débuté par la présentation d'un cahier des charges, d'un diagramme de Gantt. Le suivi du projet a ensuite eu lieu grâce à la rédaction de rapports de séance incluant tout le travail fourni entre deux séances.

En effectuant nos recherches sur Internet, principalement YouTube, nous avons pu trouver de nombreux exemples de ce qui était faisable grâce à l'Arduino. Nous étions sûrs de deux choses, nous voulions que ce soit un robot qui puisse se déplacer et que ce soit une première dans l'école. Nous avons hésité entre un bras robotisé automatisé ou un robot issu du biomimétisme.

C'est à ce moment que jaillit l'idée d'un robot chien que nous avons baptisé POLYDOG. Nous l'avons conçu dans le but de déplacer un corps à 4 pattes dans les 4 directions basique : gauche, droite, avant, arrière. Il est doté d'une PixyCam et d'un module Infrarouge pour une communication à distance des mouvements.

Ainsi, la problématique de notre sujet est : Par quels moyens déplacer un robot sur 4 pattes ?



Fig. 2 : Le PolyDog se cache dans les fourrés

Cahier des charges

Reprenons notre cahier des charges initial et faisons un point sur les objectifs que nous nous étions fixés :

CAHIER DES CHARGES FONCTIONNEL DU ROBOT

Présentation générale du projet

Inspiration

Au commencement, nous avons fait pleins de recherches sur les projets qui ont déjà été fait pour nous inspirer et avoir un aperçu de ce qui serait faisable de créer.

Nous nous sommes vite orientés vers un robot arduino qui se déplace, notamment avec un certain équilibre. On s'est alors penchés sur le robot Spot de Boston dynamics, puis sur le chien Bittle de Petoï, Spot Micro et Mini Pupper.

Objectifs du Projet

Nous souhaitons créer un robot chien, son nom est PolyDog, une structure plastique montée sur quatre pattes articulées. Il sera télécommandé par une manette xbox en bluetooth.

Fonctionnalité de PolyDog

- Marcher dans les 4 directions
- Combinaison de commande pour faire certaines actions préprogrammées du robot.
- Rotation sur lui-même

2.1/ Fonctionnalités réalisées

Nous avons quasiment rempli l'objectif principal : notre robot peut marcher en avant et sur les côtés, avec des combinaisons de commande pour faire certaines actions préprogrammées du robot.

Cependant, nous avons pour objectif qu'il soit télécommandé par une manette de Xbox. Nous ne sommes pas arrivés à ce degré de complexité, en revanche, nous avons équipé le robot de commandes à distance infrarouges.

De plus, nous n'avons pas eu la possibilité de réaliser la fonctionnalité lui permettant de tourner sur lui-même.

2.2/ Fonctionnalités optionnelles

Nous avons équipé Polydog, d'une caméra intelligente qui suit un objet d'une certaine couleur. Le module PixyCam installée sur son socle à l'avant du robot. Lorsqu'elle détecte un objet bleu, le robot avance tout droit.

2.3/ Diagramme de Gantt et Planning réel

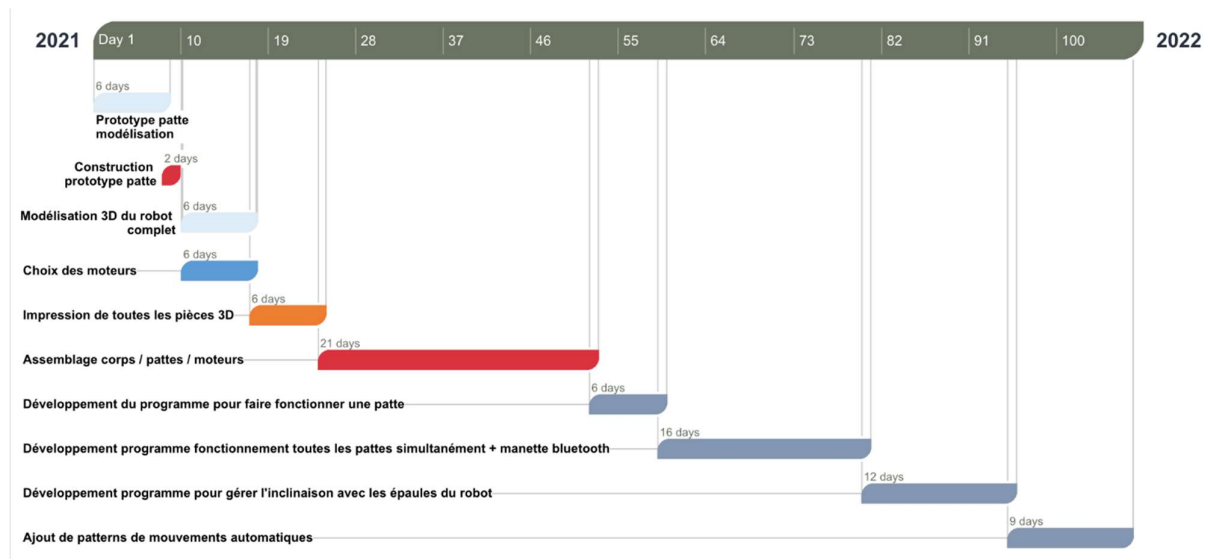


Fig. 3 : Diagramme de Gantt initial

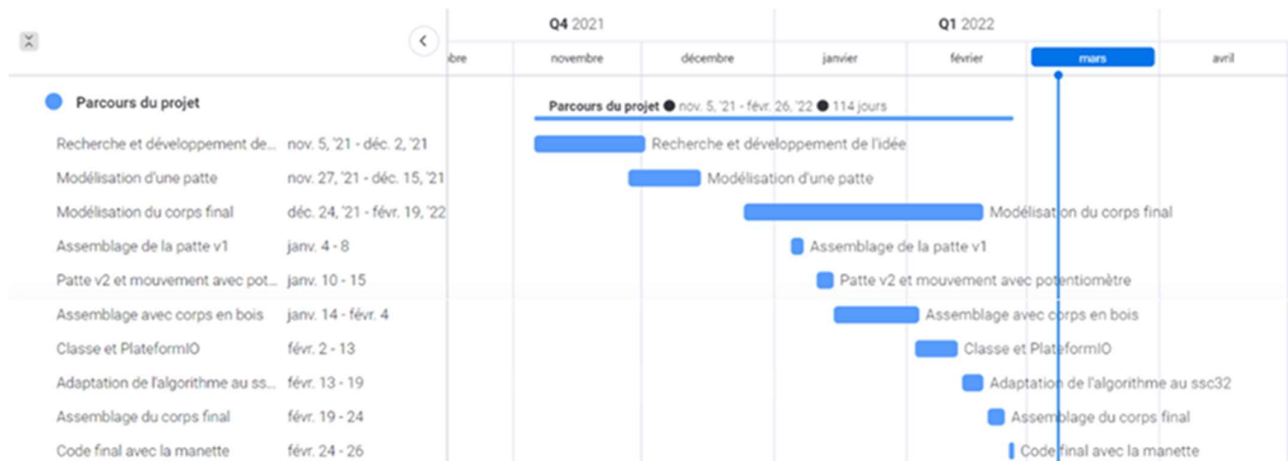


Fig. 4 : Planning réel

La modélisation 3D du robot complet a pris beaucoup plus de temps que prévu. Cela vient du fait de l'évolution de notre vision du robot et de ses composants. Il faut savoir que nous avons eu beaucoup de problème au niveau de la puissance des moteurs et de l'équilibre du robot. Pensant que le problème venait de la gestion des circuits de la carte, nous avons changé 4 fois de système électronique :

- Driver Adafruit + ArduinoNano
- ArduinoMega2560+ Base Shield + SensorShield (Empilés les uns sur les autres pour obtenir les pins adéquat)
- MegaStrong + Board micro controller intégré
- ssc32- servo controller + Arduino UNO

A chaque changement de système, nous avons l'impression de revenir en arrière du fait de devoir changer le programme et aussi l'organisation des pattes pour des raisons parfois inconnues.

Hardware

3.3/ Pièce 3D



Composition du robot :

Il mesure 17,6 cm de large sur 42,4 cm de long. Il pèse environ 2 kilogrammes.

37 pièces 3D indépendantes assemblées avec 142 vis, plus de 142 écrous et 14 inserts.

Le temps d'impression peut s'estimer à plus de 100 heures.

La conception a été réalisée sur OnShape, un logiciel en ligne, gratuit pour les étudiants.

Le boîtier pour les moteurs a été choisi circulaire pour pivoter avec plus de liberté de gauche à droite.

Nous avons rencontré des difficultés dans la conception de certaines pièces. Les côtes du bas devaient s'emboîter avec le plateau avec les moteurs déjà fixés dessus. Nous avons dû faire face également au problème du poids. Les deux pièces sternum et les côtes du bas ont été optimisées pour peser le moins possible tout en gardant une certaine solidité.












Durant l'assemblage, 2 pièces ont été endommagées. Avec un peu de colle, l'emboîtement a pu se faire sans trop de problèmes.

Cependant, le montage du robot entier en abandonnant le corps en bois l'a rendu trop lourd ce qui a réduit ses capacités de mouvement.

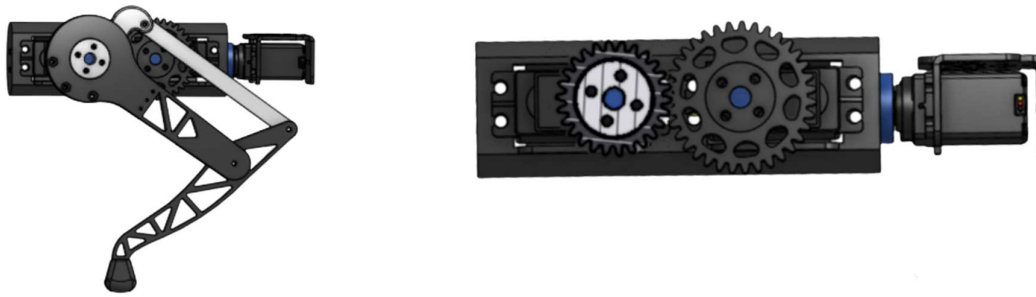
Le poids cumulé de toutes les pièces pèse sur l'équilibre et la mobilité des pattes.

Cependant, nous ne regrettons pas, car nous y avons appris beaucoup de choses : conserver une marge entre les pièces car l'impression a une imprécision de 0,4 mm ; mettre des inserts dans le plastique ; choisir minutieusement l'emplacement des vis.

Si c'était à refaire, j'opterais pour un corps encore plus léger, en dépit que les fils pendraient sur le côté et au-dessus du robot. Le rendu final serait moins ordonné mais plus mobile.

Part	Quantity	Part	Quantity	Part	Quantity
 genou_v3 - Right Part	2	 genou_v3 - Left Part	2	 cote_du_bas_v2	1
 hanche_v2 - Right Part	2	 hanche_v2 - Left Part	2	 plateau_v3	1
 femur_v2 - Right Part	2	 femur_v2 - Right Part	2	 sternum_ALUM_v4	1
 pied_v1	4	 tibia_v1	4	 sternum_USB_v5	1
 tendon_v1	4	 boitier_v2	4	 carapace_v2	1
 engrenage_v2	4				

3.2/ Fonctionnement de la patte



Nous avons opté pour un fonctionnement d'une patte avec le modèle du parallélogramme. Les moteurs sont centralisés au niveau du corps du robot pour garder la patte légère et une optimisation du mouvement.

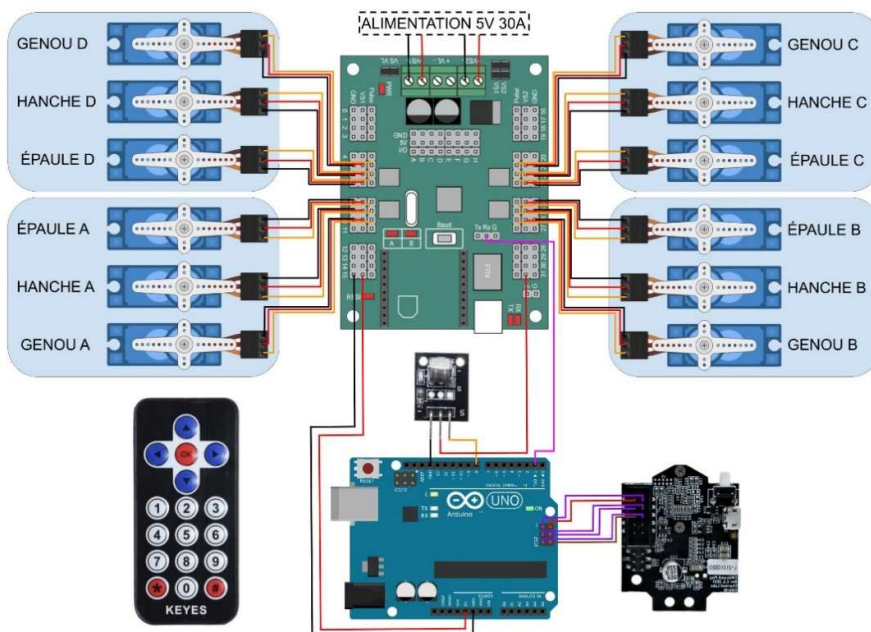
Une patte du robot est munie de 3 servos moteurs. Le 1^{er} servo en partant de la gauche est celui qui bouge le genou (au centre de la patte). Le genou est positionné de tel sorte que l'amplitude de mouvement (0° à 180°) du servo associé correspondent aux angles désirés.

Le 2^e est relié à un engrenage de 35 dents vers 26 dents qui bouge la hanche. De ce fait le rapport entre les deux engrenages augmente l'amplitude de mouvement de la hanche de 1,35, pour obtenir 0° minimal jusqu'à 242° maximal.

Le 3^e est fixé au corps principal du robot et permet de faire pivoter la patte sur le côté. Pour soutenir le boîtier, on place une fixation pivot à l'autre extrémité.

Le pied du robot est fait en plastique élastique caoutchouteux. Cela permet d'obtenir une plus grande adhérence au sol.

3.3/ Composants électroniques



Nous avons ici, l'ensemble des composants électroniques utilisés pour notre projet. Nous avons 12 servo moteurs branchés sur un ssc32-servo controller qui communique par l'intermédiaire du Serial avec une Arduino UNO. Cette dernière est le composant central puisqu'elle relie sur ses ports, la PixyCam et le récepteur infrarouge. L'ensemble est alimenté sur secteur par l'intermédiaire d'une alimentation de 5V et 30A.

Software

4.1/ Classes du programme

La partie code de notre projet a été implantée sur un espace de travail PlatformIO. Il permet de créer un code source unique pour de multiples environnements facilement (Arduino Uno, Mega 2560, etc). Dans le dossier source, nous avons 3 classes :

- CustomServos :

Cette classe permet de communiquer avec le SSC32 en indiquant seulement un angle. On instancie un objet de cette classe à partir d'un numéro de pin.

```
void CustomServo::move(int position, int time)
{
    Serial.print("#");
    Serial.print(_pin);
    Serial.print(" P");
    Serial.print(position);
    Serial.print(" T");
    Serial.println(time);
}
void CustomServo::write(int angle)
{
    move(map(angle, 0, 180, 500, 2500), 0);
}
```

Par l'intermédiaire de la carte Arduino, on envoie des commandes sur le Serial du ssc-32. Le message envoyé est de la forme « #{numéro_pin} P {position entre 500 et 2500} T {time}

Nous n'utilisons pas le paramètre temps dans la fonction move. Cela nous permet d'avoir la liberté de faire varier le temps dans les méthodes codant les mouvements du robot.

La fonction write renvoie la fonction move en convertissant l'amplitude 500-2500° en 0-180°. Cela facilite la compréhension physique des mouvements du robot.

- Leg :

Permet de contrôler chaque moteur d'une jambe sans se soucier des décalages. Par exemple, les moteurs de la hanche n'ont pas les mêmes orientations entre la gauche et la droite. Pour permettre une synchronisation des moteurs des 4 pattes, nous avons pour certains, ajouter un offset de 170°.

- PolyDog :

Cette classe est en quelque sorte un répertoire des méthodes qui vont permettre de donner vie à notre chien. On y définit la méthode self_balancing(), hold_shoulder(), move_forward(), move_right(), move_left(), etc

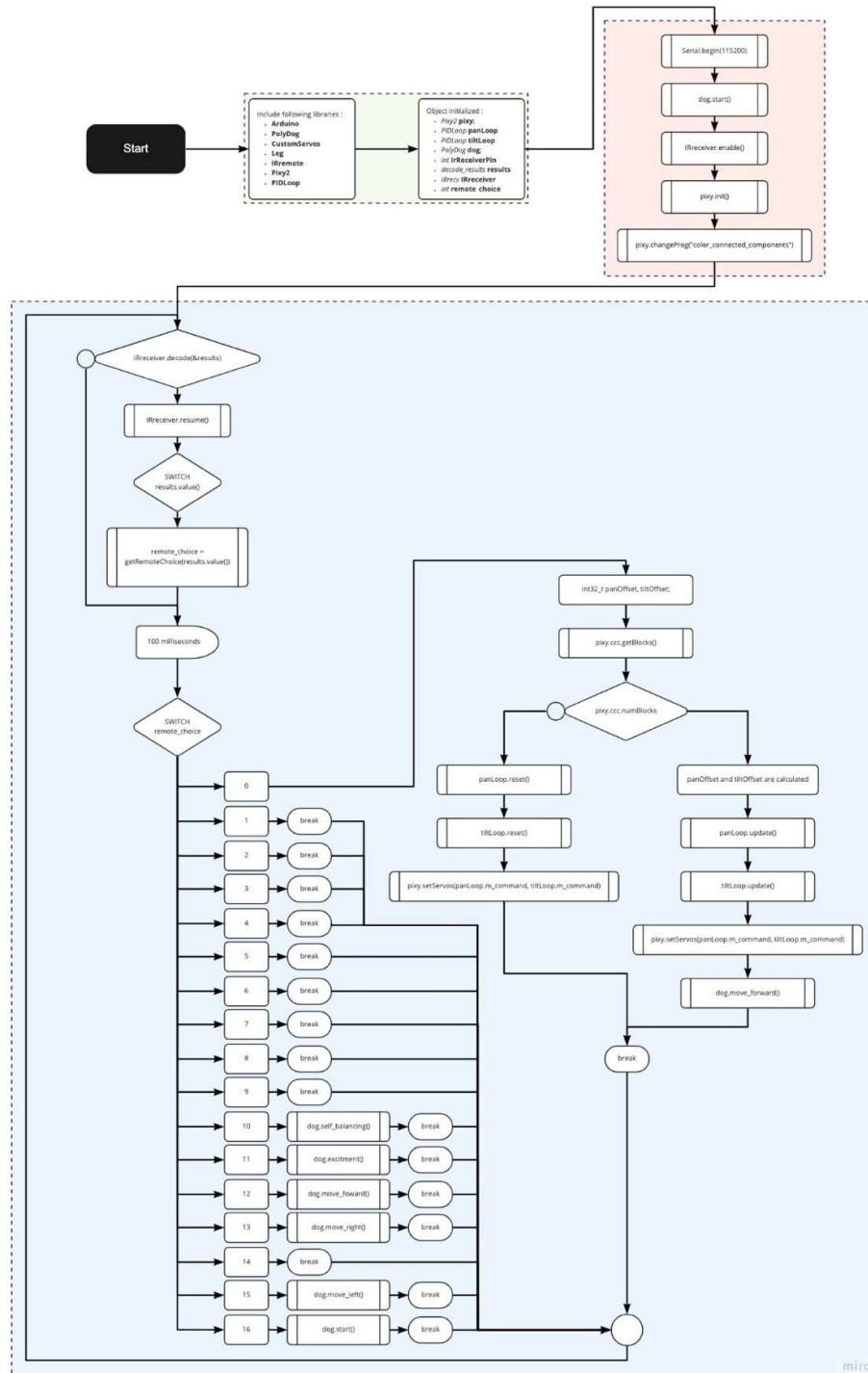
Les angles de chacun des servos moteurs sont choisis à partir de nombreux test d'optimisation du mouvement de la patte.

```
void PolyDog::hold_shoulders()
{
    legA.move_shoulder(90);
    legB.move_shoulder(90);
    legC.move_shoulder(90);
    legD.move_shoulder(90);
}
```

Voici un exemple simple d'une méthode dans la classe Polydog. Cette méthode maintient les pattes perpendiculaires au sol.

On remarque que cette méthode utilise la classe Leg pour gérer les décalages dû à notre installation des servo moteurs.

4.2/ Algorithme



Les librairies et modules sont d'abord importés (partie verte). Ensuite, les objets utilisés sont instanciés au début du programme (partie rouge). C'est à ce moment-là que le robot est allumé ainsi que la PixyCam et la communication infrarouge. Le robot est alors debout. Enfin, une boucle détectant l'appui des boutons de la télécommande permet de choisir les actions du robot. Par exemple, si l'on appuie sur le bouton 0, le robot utilise la PixyCam pour détecter s'il y a un objet bleu. Si c'est le cas, il se dirige vers l'avant, sinon il se maintient debout. Les autres boutons lui permettent d'effectuer un enchainement de mouvement en boucle ou de se déplacer.

Conclusion

Ce projet Arduino était motivant et nous a apporté beaucoup de nouvelles compétences. Nous avons appris que le temps de réalisation d'une tâche est très variable. Nous nous sommes formés sur l'utilisation de nouveaux outils et de nouveaux logiciels. Notre perception du projet a évolué au fur et à mesure de notre gain d'expérience.

C'est une belle aventure qui nous a fait découvrir le monde de la robotique. Cela a conforté Hugo dans son choix d'aller en robotique l'année prochaine et a conforté Raphaël dans son choix d'aller en SI.

Perspective

Certes, notre robot chien Polydog marche et est télécommandé. Nous avons pu placer une caméra intelligente qui n'était pas prévu au départ. Nous avons appris de beaucoup de nos erreurs et surmonter des difficultés telles que la non-réponse des servo-moteurs, des angles incohérents, des problèmes de communication entre cartes Arduino, etc...

Cependant, le résultat n'est pas réellement convaincant. Il avance et se déplace en général d'une distance très faible (1cm par seconde en moyenne). Pour certains mouvements, il faut le soutenir un petit peu pour l'alléger artificiellement.

C'est pourquoi, nous considérons ce robot, comme la version 1, le premier jet. Nous prévoyons de commencer une nouvelle version avec un corps en aluminium ou impression résine et de meilleurs servo moteurs, et surtout l'expérience acquise au cours de 3 mois de travail. Nous avons une dizaine de semaine pour le finir et nous sommes plus motivés que jamais pour y arriver.

Bibliographie

Inspiration :

https://www.youtube.com/watch?v=3_MtMmjEwa4&t=180s

<https://www.yeggi.com/q/boston+dynamics/>

<https://grabcad.com/library/walking-quadruped-robot-diy-1>

Lien guide Lynxmotion ssc32-servo controller :

http://www.lynxmotion.com/images/data/lynxmotion_ssc-32u_usb_user_guide.pdf

Guide de mise en marche de la télécommande IR :

<https://www.gotronic.fr/pj2-36228-2302.pdf>

<https://www.gotronic.fr/pj2-codes-gt017-2234.pdf>

Notre chaîne YouTube :

<https://www.youtube.com/channel/UCp-MOXWuPocJWpOKq-EK11g/featured>

