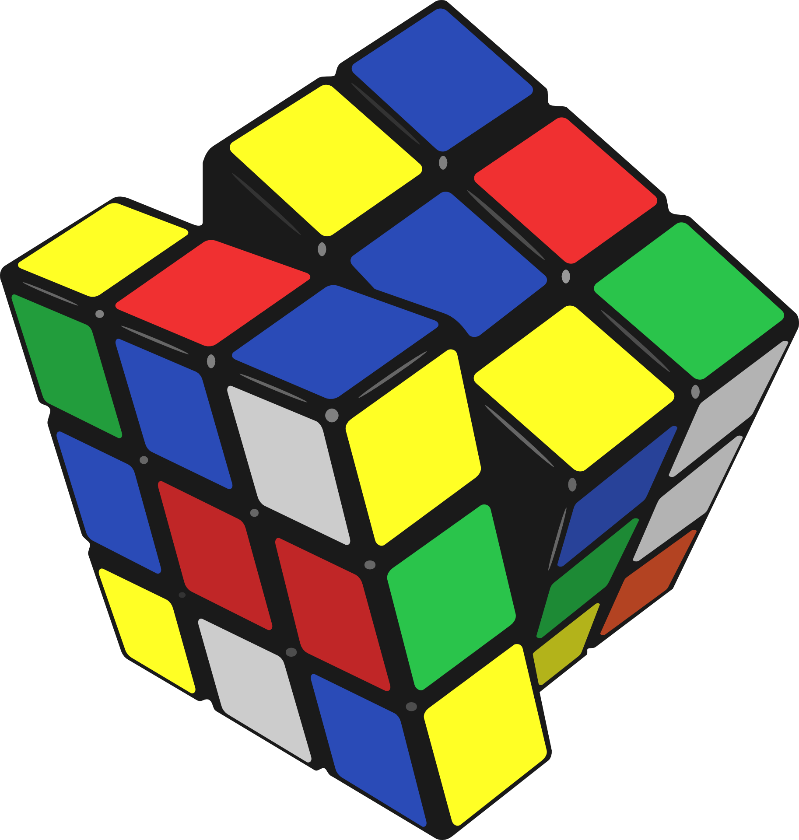
Rubik’s Solver





Résolveur automatique de Rubik’s Cube

LabVIEW NI Vision EAGLE SolidWorks Cura CorelDRAW

Table des matières

[Introduction 3](#_Toc73093415)

[Planning et diagramme de Gantt 4](#_Toc73093416)

[Documents du dossier technique 5](#_Toc73093417)

[Manuel d’utilisation 7](#_Toc73093418)

[Communiqué de presse 11](#_Toc73093419)

[Algorithme de résolution 12](#_Toc73093420)

[Synoptique du projet de résolution LabVIEW 14](#_Toc73093421)

[Difficultés rencontrées 15](#_Toc73093422)

[Conclusion 16](#_Toc73093423)

# Introduction

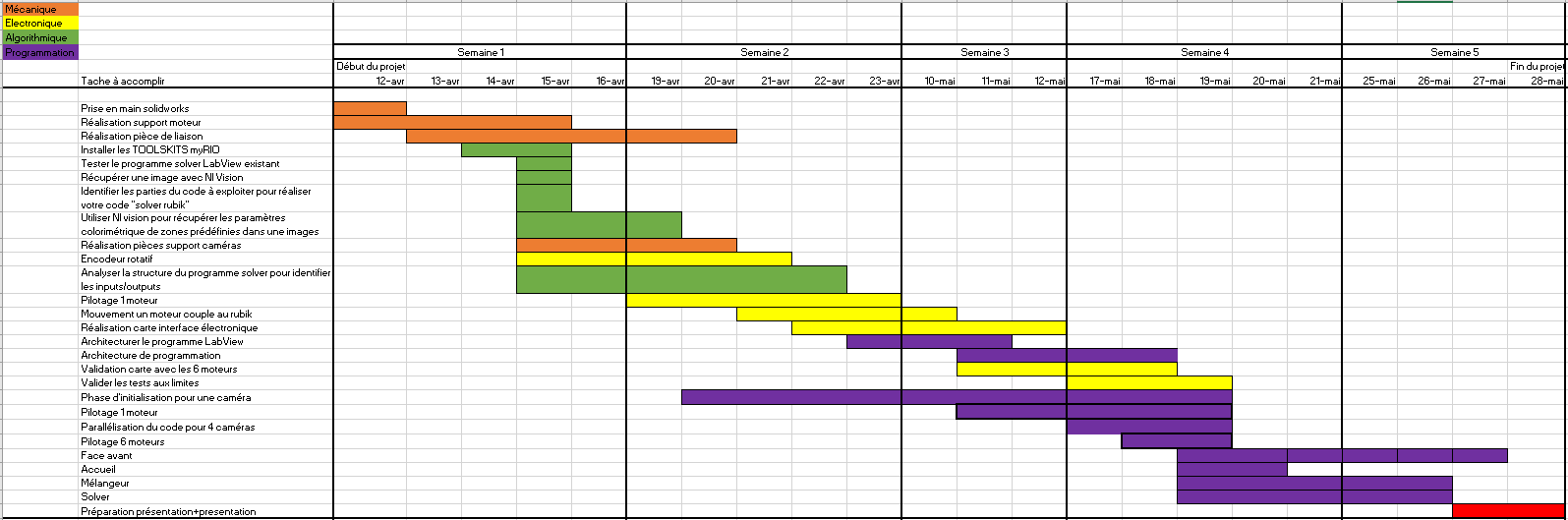
Dans le cadre de notre deuxième année en école d’ingénieurs à l’ESIGELEC dans le département Systèmes Embarqués et Ingénierie, Elodie, Kerrian, Juliette et moi avons choisi pour notre projet S8 le projet Rubik’s Cube. Nous venons tous de dominantes différentes : ISE-VA, ISYMED, MCTGE et ISE-OC. Les compétences et les connaissances que nous avons apportées au projet sont donc variées et viennent directement de ce qui nous a été enseigné dans nos dominantes.

Pour ce projet nous avons donc dû imaginer, créer et modéliser plusieurs pièces et parties du projet final, que nous avons donc ensuite assembler en un projet fonctionnel. L’apprentissage de nouveaux logiciels et l’approfondissement de ceux que nous connaissions déjà a fait partie intégrante du projet pour tous les membres du groupe.

Nous avons découpé les tâches entre les membres du groupe, selon nos compétences respectives et notre connaissance des logiciels nécessaires. Elodie s’est occupée de la partie documentation et d’une partie de l’acquisition des images via les caméras. Kerrian s’est quant à lui occupé de l’algorithme d’acquisition des images et de la transformation des informations récoltées en données exploitables et d’autres VIs nécessaires à la résolution du cube. Juliette s’est penchée sur le fonctionnement des drivers, des moteurs et des démultiplexeurs afin de faire fonctionner correctement le système électronique, ainsi que sur les VIs permettant de faire tourner les faces du cube. Enfin, Lucas s’est chargé de la confection de la carte électronique sous EAGLE et de la réalisation de celle-ci, et également sur les VIs correspondant aux interfaces utilisateur et ceux correspondant à certaines fonctionnalités.

Nous avions cinq semaines pour réaliser ce projet, sachant que les seules parties du projet qui nous étaient fournies au début de celui-ci sont la structure en bois qui accueille le cube, trois des quatre pieds ainsi que le support pour le moteur du haut. Pour la partie algorithmique et logicielle, le projet de résolution du cube nous était également fourni.

# Planning et diagramme de Gantt



Pour ce projet, nous nous sommes répartis les tâches entre tous les membres et nous nous sommes fixé des dates limites afin d’être sûrs de pouvoir rendre le projet dans les temps. Cette stratégie s’est avérée payante.

Lorsque chaque partie du projet était terminée, nous l’intégrions au projet complet afin de s’assurer de sa compatibilité. Ainsi lorsque nous avons assemblé toutes les pièces ensembles à la quatrième semaine, nous n’avons pas eu beaucoup de problèmes d’intégration ou de compatibilité.

# Documents du dossier technique

* **Fichiers CAO SolidWorks et STL:**
* *Pièce de liaison. SLDPRT* : schéma SolidWorks de la pièce de liaison entre le moteur et le Rubik’s cube
* *Pièce de liaison. STL* : vue 3D de la pièce de liaison
* *Support moteur. SLDPRT* : schéma SolidWorks de la pièce support moteur qui sert à porter le moteur
* *Support moteur. STL* : vue 3D de la pièce support moteur
* *BoitemyRIO.SLDRT* : schéma SolidWorks de la boîte contenant le myRIO et la carte électronique.
* *BoitemyRIO.STL* : vue 3D de la boîte.
* *Couvercle.SLDRT :* schéma SolidWorks du couvercle de la boîte.
* *Couvercle.STL* : vue 3D du couvercle de la boîte.
* **Fichiers Eagle :**
* *Branchements myRIO.xlsx :* Fichier Excel présentant le branchement entre la carte électronique et le myRIO.
* *Driver Board.docx :* Fichier présentant le branchement des drivers
* *myRIO.docx :* Fichier présentant les branchements à faire sur le myRIO pour le bon fonctionnement de la carte.
* *DriverOK-DemuxOK-VIA71.brd :* fichier nécessaire pour la création de la carte. Présente le placement et la répartition des différentes pièces.
* *DriverOK-DemuxOK-VIA71.sch :* schéma des branchements nécessaires pour le bon fonctionnement de la carte.

* **Datasheets :**
* *Algorithme de résolution.docx*: Algorithme qui montre le schéma à suivre pour la résolution du Rubik’s cube
* *Tuto install myrio.pdf* : Fiche expliquant comment installer les extensions nécessaires au myRIO.
* *demultiplexer\_hcf4051.pdf* : Datasheet des démultiplexers utilisés.
* *pj-1164.pdf*: Datasheet des moteurs pas à pas utilisés.
* *StepperMotorDriver\_A4988.pdf* : Datasheet des drivers utilisés.
* *HKT22.pdf* : Datasheet de l’encodeur utilisé.
* **Projet LabVIEW :**
* *Synoptique de l’architecture du programme LabVIEW.docx* : schéma simplifié du programme de résolution LabVIEW
* *Diagramme de Gantt.xlsx*: Diagramme donnant l’organisation du projet
* *Manuel d’utilisation.docx* : Fichier donnant les étapes à suivre afin d’avoir son Rubik’s Solver
* *Documents du dossier techniques.docx*: ce document
* *Communiqué de presse.docx*: Document en anglais qui donne une explication détaillée de ce en quoi consiste le projet et son organisation

Légende :

**En gras** : les dossiers

*En italique* : les fichiers

# Manuel d’utilisation

Matériel nécessaire :

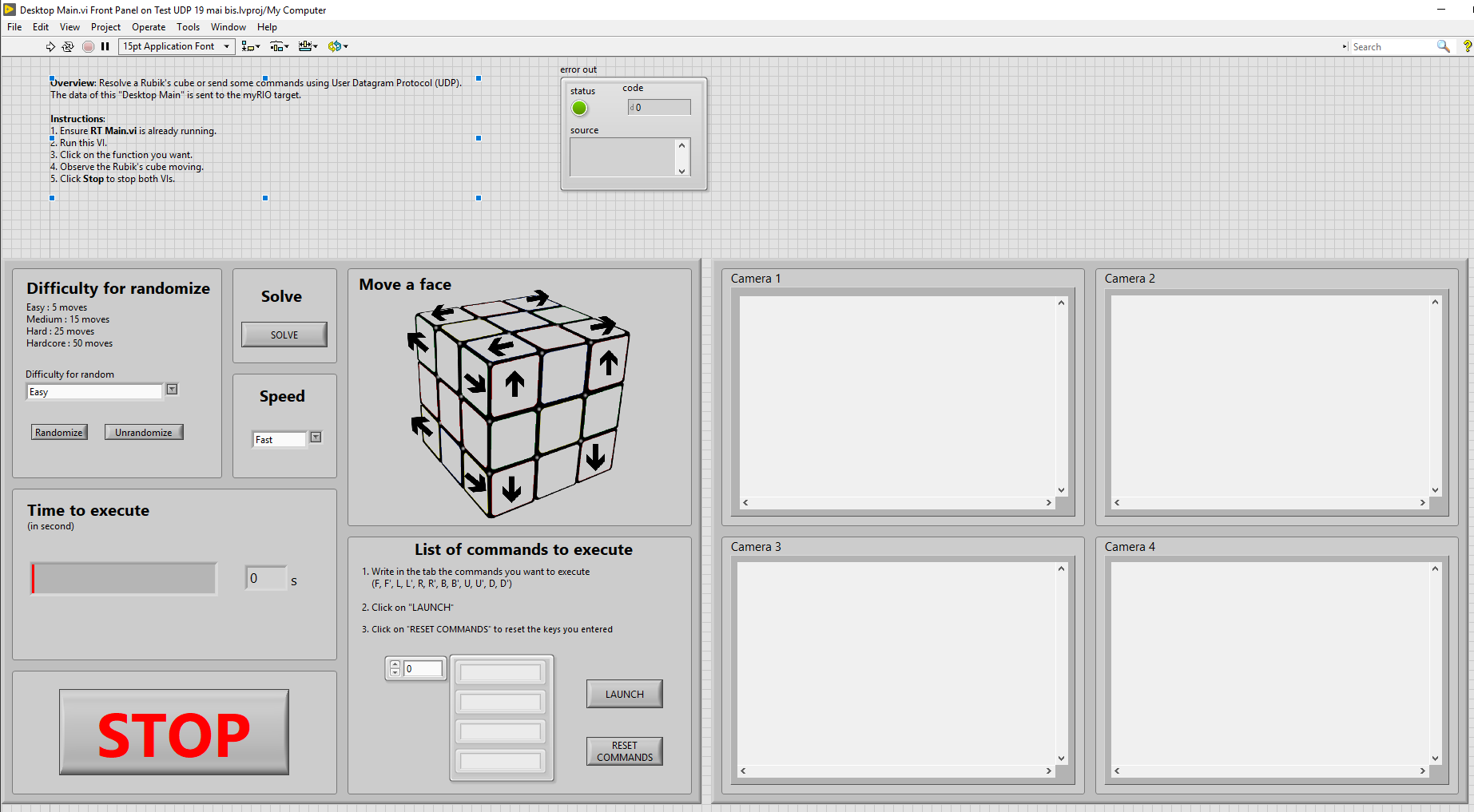
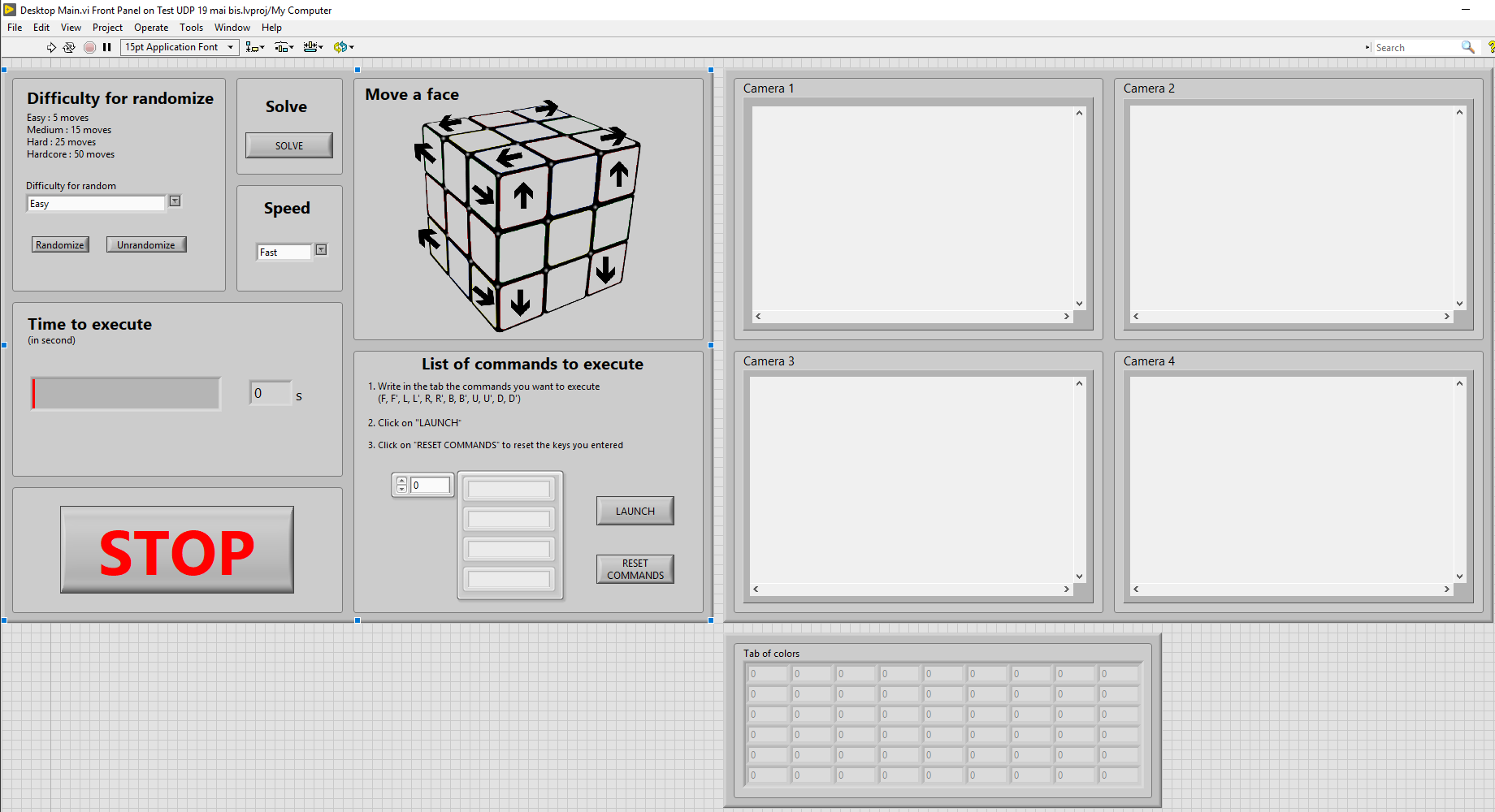
* Une imprimante 3D
* Un kit Rubik’s Solver contenant : 4 caméras, 6 moteurs pas à pas, un myRIO, une structure Rubik’s Solver, 4 supports moteurs, 6 Drivers et un Rubik’s cube, 2 démultiplexers.

Afin de construire le Rubik’s Solver vous devez :

* Installer le logiciel myRio sur votre ordinateur. Pour cela vous avez le « *Tuto install myrio.pdf* » pour vous aider, présent dans le dossier « **Datasheets** ».
* Imprimer le pied manquant, servant de support au moteur, grâce au fichier « *Support moteur. SLDPRT* » présent dans le dossier « **Fichiers CAO SolidWorks et STL »**.
* Imprimer les six pièces de liaisons entre les moteurs et le Rubik’s cube grâce au fichier « *Pièce de liaison. SLDPRT* » présent dans le dossier « **Fichiers CAO SolidWorks et STL »**.
* Créer la carte électronique à l’aide des patrons : « *Eagle* » présent dans le dossier « **Fichiers Eagles**».
* Faites le montage de la machine :
* Placez le pied, support moteur, à son emplacement
* Attachez les pièces de liaisons aux moteurs
* Placez les moteurs sur leurs supports
* Placez le Rubik’s cube au centre de la structure
* Branchez la carte électronique comme indiqué dans le fichier « *Branchements myRIO.xlsx*» dans le fichier « *Branchements* ».
* Téléchargez et ouvrez le projet LabVIEW.

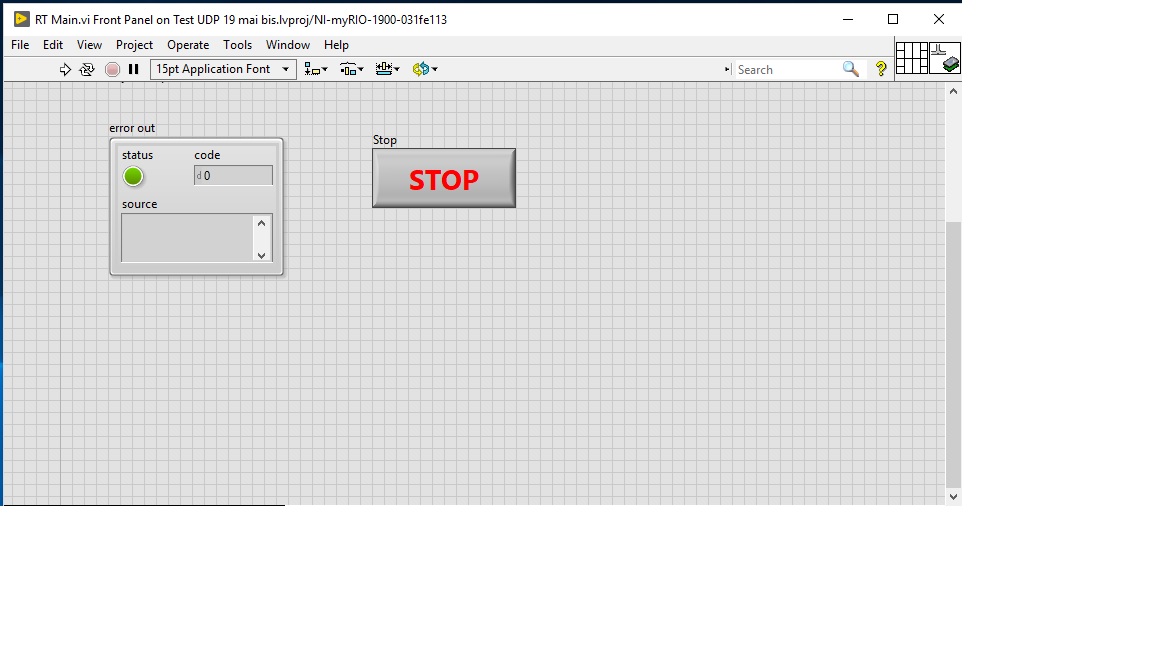
Pour faire fonctionner le Rubik’s Solver vous devez :

* Ouvrez les VIs : RT Main.vi et Desktop Main.vi
* Lance d’abord le vi RT Main.vi puis le vi Desktop Main.vi
* Ceci est la face avant du vi Desktop Main.vi:



* Les différentes actions possibles sont :
* « **Randomize** » afin de mélanger le Rubik’s cube. Il y a différents niveaux de difficulté :
* Easy : le programme fait tourner aléatoirement 5 faces du Rubik’s cube
* Médium : le programme fait tourner aléatoirement 15 faces du Rubik’s cube
* Hard : le programme fait tourner aléatoirement 25 faces du Rubik’s cube
* Hardcore : le programme fait tourner aléatoirement 50 faces du Rubik’s cube
* « **Unrandomize** » afin de résoudre le Rubik’s en faisant les mouvements inverse que ceux pour le mélanger.
* « **Solve** » pour résoudre le Rubik’s Cube. Vous allez voir un patron du Rubik’s cube apparaitre avec les couleurs des facettes, vous devez vérifier si les couleurs correspondent bien à la réalité puis cliquer sur « Valider les couleurs ».
* « **Stop** » afin d’arrêter le vi.
* « **Speed** » pour choisir la vitesse de rotation :
* Slow : Lent
* Medium : Vitesse moyenne
* Fast : Rapide
* Une image du Rubik’s cube avec des flèches qui permet de tourner la face que l’on souhaite en appuyant sur la flèche.
* « **Launch** » permet d’entrer manuellement une liste de commande tel que (voir lettre sur le dos des moteurs):
* F / F’ : Tourne la face de devant dans un sens ou un autre
* R / R’ : Tourne la de droite avant dans un sens ou un autre
* L / L’: Tourne la face de gauche dans un sens ou un autre
* B / B’ : Tourne la face de derrière dans un sens ou un autre
* U / U’ : Tourne la face du dessus dans un sens ou un autre
* D / D’ : Tourne la face du dessous dans un sens ou un autre

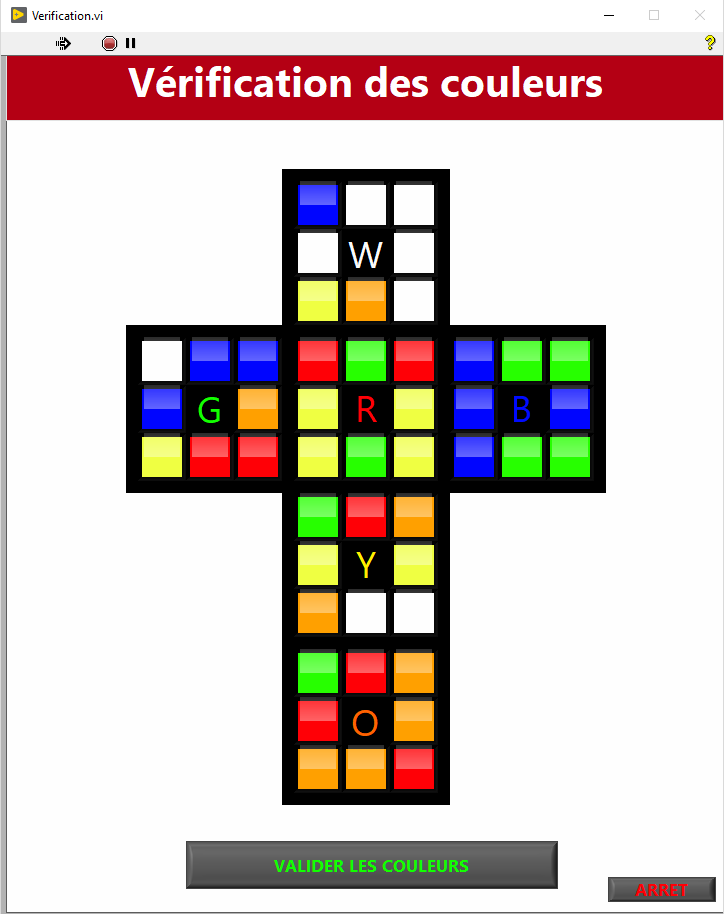
Sur la face avant de RT Main.vi, vous pouvez arrêter le programme en cliquant sur le bouton **STOP**.



Lorsque le bouton **Solve** est appuyé, la face avant du vi « *Verification.vi* » s’affiche.

Si l’un des carrés n’est pas de la bonne couleur, il suffit de cliquer dessus pour en changer sa couleur.

Exemple de patron du Rubik’s Cube que vous pouvez obtenir :



# Communiqué de presse

This project consists in the build and the coding of a machine which will be used to randomize and solve a Rubik’s Cube. This will be useful for people who don’t know how to solve a cube and to see how a cube can be solved.

The goals for this projects are the following:

- Imagine and create the parts holding the sixth motor and the parts linking the motors to the cube using SolidWorks software.

- Imagine and create an electronic board to hold the motor drivers and the demultiplexers which will be used to control the motors and therefore solving the cube. We created the board using the Autodesk EAGLE software.

- Develop a LabVIEW program on real-time target NI myRIO which will be used to randomize and solve the cube by generating signals and PWMs and analyzing the cameras’ data, allowing the program to determine which color is each part of the cube.

To realize this project, we are a 4-people team. The tasks are separated between each member of the team. Lucas took care of the creation of the electronic board and welded it correctly, Juliette analyzed the way the motor are working and managed to make them work by creating new VIs, Kerrian was in charge of the analysis of the colors of the pictures given back by the cameras using machine learning and finally, Elodie dealt with the analysis of the existing LabVIEW project which solves a cube in whatever configuration it is.

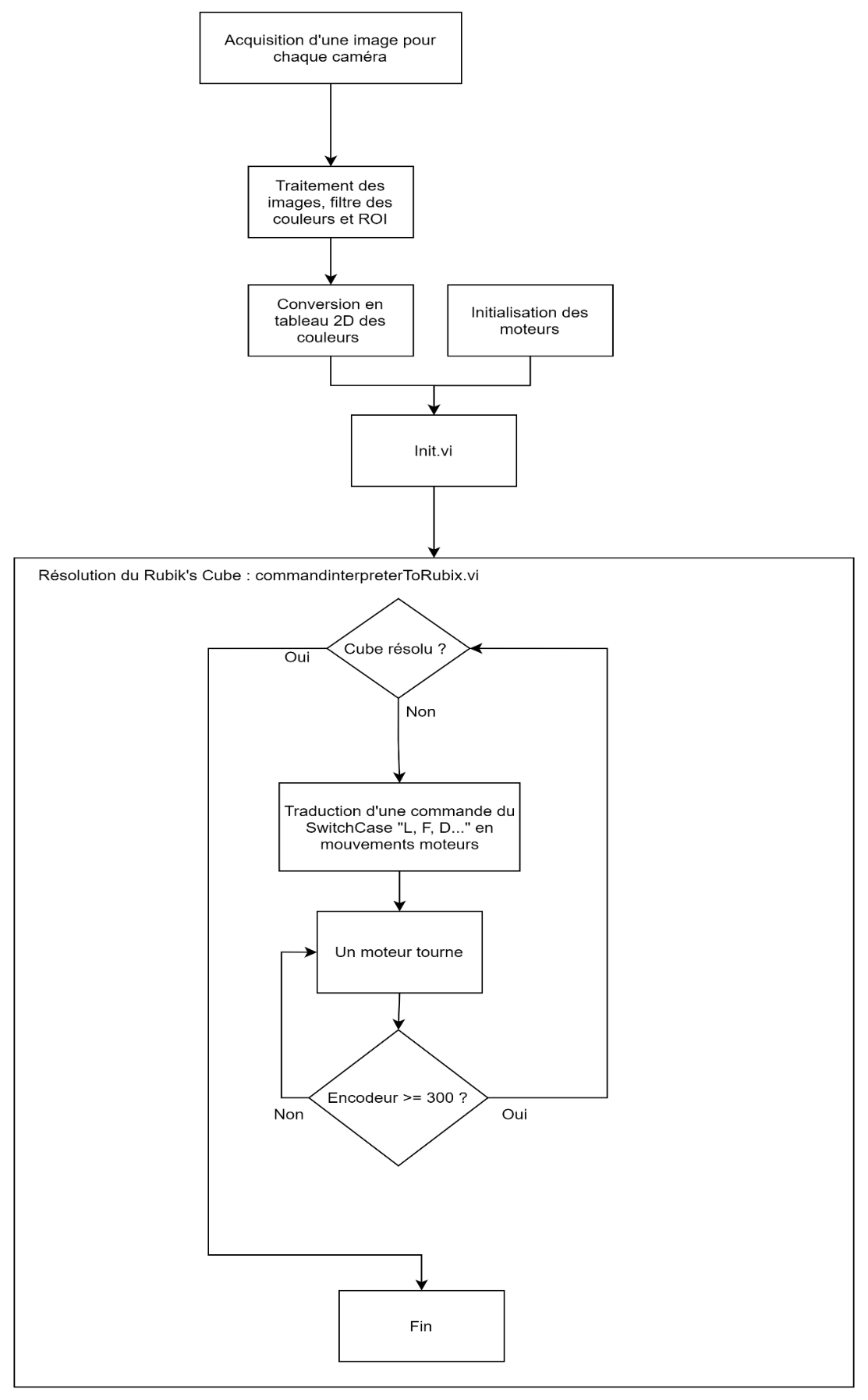


For this project we used different technologies and software i.e. we used SolidWorks and a 3D printer to create various spare parts of our machine. Other software such as EAGLE and LabVIEW were useful to develop answers to the problems we were facing, such as the electronic board and issues with the welding or the reflection of the light on the cube.

Since the beginning of this project we already have created the missing spare parts, fixed the cameras to the stand, created and fixed the electronic board, created a VI allowing us to rotate the motor of exactly 90° and finally we created VIs analyzing the colors sent back by the cameras. Lastly, we created a VI allowing us to control the motors depending on the string command the tester entered. This principle will be used during the solving of the cube, to command one or the other motor in the order given by the solving LabVIEW project.

Therefore, the electronic board and the Vis allowing us to control the motors are done and ready to be used

# Algorithme de résolution



Algorithme du projet de résolution du Rubik’s Cube

Le schéma ci-dessus montre le cheminement du projet nous permettant de résoudre un Rubik’s Cube à l’aide de moteurs.

On commence avec l’acquisition des images par les caméras, qui sont traitées par le logiciel et desquelles on extrait les couleurs des différentes facettes de chaque face du cube grâce à des ROIs. Ces couleurs sont alors arrangées dans un tableau 2D et le programme de résolution du Rubik’s Cube se lance alors. On initialise à ce moment-là les moteurs et on entre alors dans une boucle qui ne se terminera que lorsque toutes les commandes nécessaires à la résolution du cube seront exécutées. Pour la précision, une valeur de 300 pour les encodeurs est équivalente à un quart de tour.

# Synoptique du projet de résolution LabVIEW

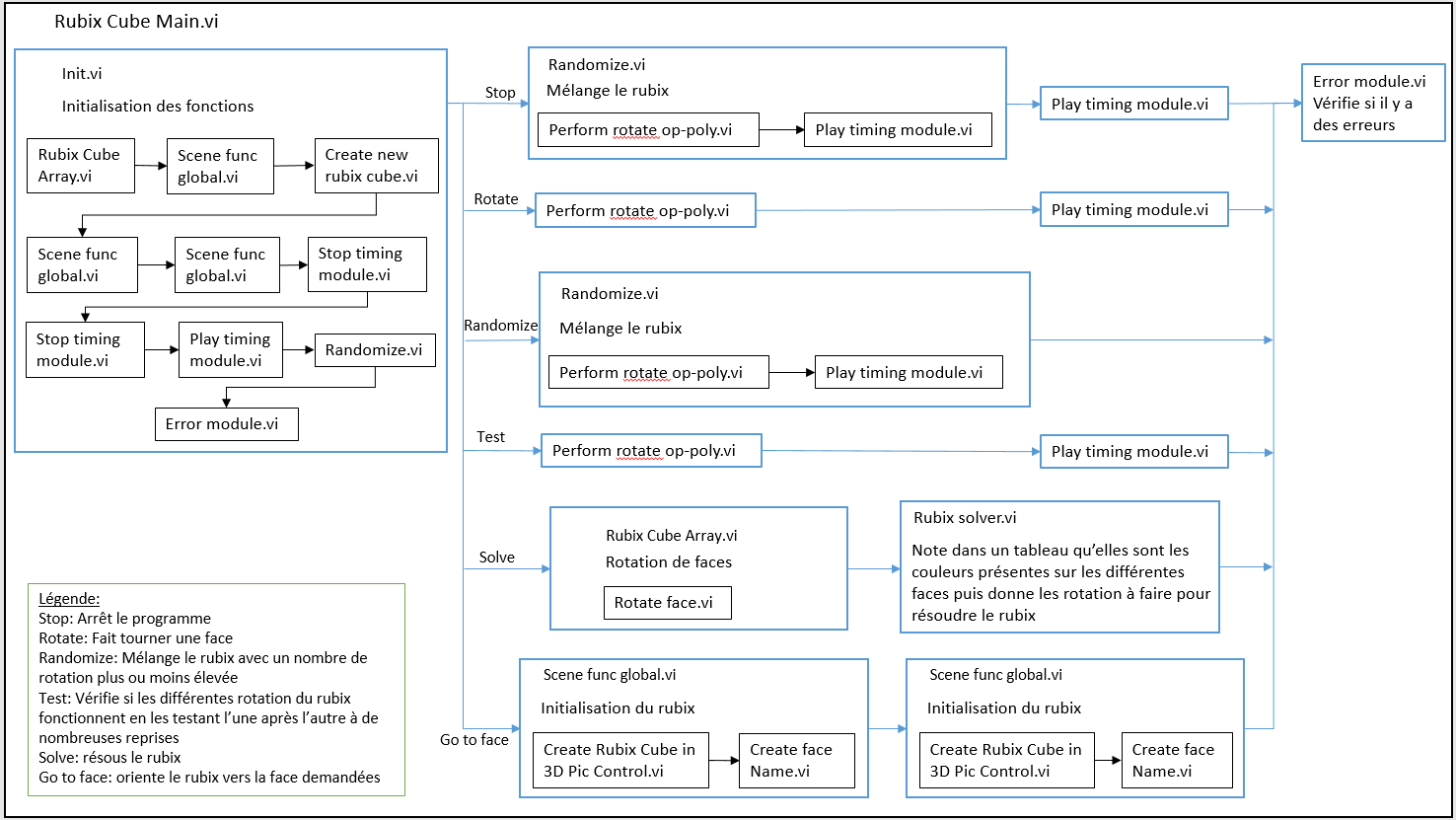


Schéma montrant le cheminement du projet de résolution du Rubik’s Cube (projet LabVIEW fourni au début du projet)

Ce schéma représente le fonctionnement du projet Rubik’s Solver qui nous permet de connaitre les mouvements à effectuer afin de résoudre le Rubik’s Cube. Après l’initialisation de la scène 3D qui permet de visualiser un cube virtuel, il est possible d’exécuter plusieurs commandes différentes, comme par exemple de tourner les faces une à une ou de mélanger aléatoirement le cube ou bien sûr de le résoudre.

C’est de ce programme que nous extrayons la solution pour nos Rubik’s Cube, et donc que nous connaissons l’ordre et le sens des moteurs à faire tourner.

# Difficultés rencontrées

Pour la création de la carte électronique, il a fallu apprendre à utiliser le logiciel EAGLE qui était nouveau pour tous les membres du groupe. Il a ensuite fallu souder et connecter les ports de la carte, qui comporte de nombreux vias et donc de nombreux points d’erreurs potentiels. Nous n’y avons pas échappé et nous avons donc dû résoudre les problèmes de continuité sur chacun des vias. Une fois fait, il a fallu connecter la carte et brancher les drivers et les moteurs afin de les faire tourner. Cette étape présentait quelques problèmes d’intégration et de fonctionnement, qu’il a fallu résoudre, notamment concernant l’alimentation des démultiplexeurs.

# Conclusion

En définitive, nous pouvons tous dire que ce projet nous a apporté beaucoup de connaissances dans des domaines pluridisciplinaires. Nous avons appris à maitriser de nouveaux logiciels comme EAGLE par exemple mais nous avons également améliorer notre maitrise de certains logiciels et notamment de LabVIEW. Nous avons aussi progressé dans notre capacité à travailler en équipe et dans des logiciels non techniques comme Trello et GitHub.

En utilisant de l’acquisition de couleurs, en contrôlant des moteurs pas-à-pas, en créant une carte électronique et en livrant les documents, tout en respectant le cahier des charges, nous avons réussi à terminer ce projet consistant en la résolution automatique d’un Rubik’s Cube.