

Annex

Annex A. Estudi del número de moviments

Apart de poder resoldre el cub, una altra cosa que volia aconseguir amb el Python, és poder fer una miqueta d'estudi sobre com funciona el programa de bé.

Utilitzant la opció de cultiu de dades, es pot desmuntar al atzar el cub i tornar a muntar les vegades que vulguis i de manera automàtica. No cal dir-li res al programa, només al principi quantes vegades es vol que es faci aquest procés. Al activar-se, s'obre la opció 3 del menú (desordenar al atzar) i la 5 alternativament i de manera automàtica. Això em va verificar que en principi el programa no falla, ja que he resolt el cub més de 50.000 vegades i no m'ha donat cap classe de problema.

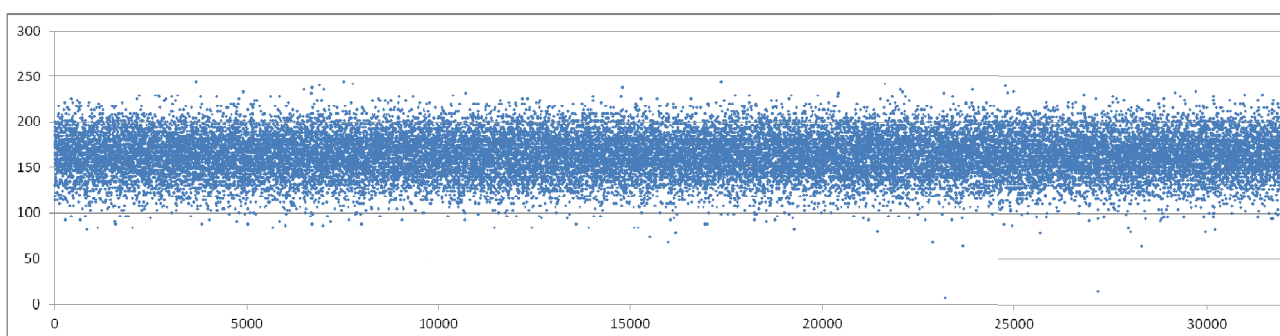
A demés, aquesta opció, compta el número de moviments que es fan per resoldre el cub en cadascuna de les vegades i les guarda en l'arxiu "Número_moviments.txt".

Així és com es veu quan s'acciona aquesta opció:

```
Mode cultiu a punt de començar  
Quantes vegades vols executar-ho?  
Possibles cultius: 5, 10, 20, 50, 100, 200, 500, 1000, 5000, 14000. 0 per sortir  
del mode cultiu
```

Quan comença l'únic que es veu en pantalla és el número de resolucions que porta que va canviant a gran velocitat. I quan arriba al número programat, torna al menú principal del programa.

Utilitzant aquesta eina, vaig obtenir un document més de 50.000 números. Amb aquests número vaig fer una gràfica, per veure quin és el mínim i el màxim possible de moviments per resoldre el cub, i quins números són més comuns.



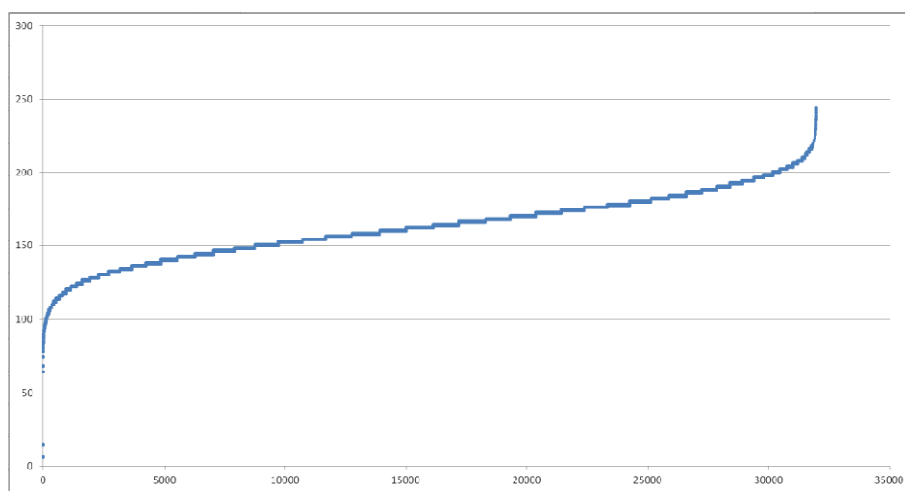
L'Excell admet com a màxim 32.000 dades en una gràfica, així que vaig arribar al màxim i no em va deixar posar més. Igualment, ja tenim una molt bona gràfica.

D'entrada veiem que cap punt passa de 250 i que la majoria estan entre 110 i 210 més o menys. Entre 80 i 110 també hi ha bastants però no tants, igual que entre 210 i 240. També podem

apreciar la improbabilitat estadística de que et surti per casualitat un estat pràcticament resolt que necessiti només menys de 50 moviments. D'entre 32000 punts, només 2 estan per sota de 50.

D'aquesta selecció de 32000 mostres, la mitjana és 162,76681. Tenint en compte que la mitjana de moviments requerits pel mètode de principiants ronda els 160, podríem dir que el programa està bastant ben optimitzat, no fa molts moviments innecessàries, i és bastant fidel a aquest mètode. El valor més alt és 244, i el més petit és 6.

També vaig crear una altra gràfica. Les dades eren les mateixes, però en aquest cas estaven ordenades de menor a major.



El que podem interpretar d'aquesta gràfica és la quantitat de resultats que hi ha hagut per cada nombre de moviments aleatoris al desmuntar. Com més petit és el pendent, és a dir, més “plana” està la gràfica, significa que hi ha més mostres que utilitzen aquest nombre de moviments. I com més pendent té, menys resultats hi ha amb aquell nombre de moviments.

Una característica de la gràfica que em va sobtar una mica al principi, es que la línia no es contínua i regular, sinó que fa com una escala. Al principi pensava que era degut a que havia masses dades i no carregava bé. Però després em vaig adonar de que no, simplement és així perquè només hi ha números enters a la gràfica. Al no haver decimals, fa un “saltet” cada cop que canvia de número.

En conclusió, el programa utilitza el mètode de principiants i resol el cub en un mínim teòric de 6 (encara que segurament, tot i que molt improbable pugui ser encara menor) i un màxim teòric de 244. El més probable, és que el resolgui entre 140 i 190 més o menys.

Annex B. Mètode Fridrich

El mètode de resolució del cub de Rubik Fridrich és un mètode creat per la enginyera elèctrica txecoslovaca Jessica Fridrich. El que aconsegueix aquest mètode és reduir significativament el número de moviments, empleats en la resolució; això sí, hi ha molts més algoritmes, i més específics i complexos. El procediment és el mateix que al mètode de principiants, comença per una cara, completa el primer nivell, el segon i finalment la última cara. Però respecte a aquest mètode, el mètode Fridrich presenta una àmpliament major dificultat i varietat d'algoritmes.

Té 3 parts: First 2 Layers (F2L), Orientation of the Last Layer (OLL) i Permutation of the Last Layer (PLL). En les següents imatges es mostren els algoritmes de cada fase.



Annex C. Com instal·lar mòduls al Python

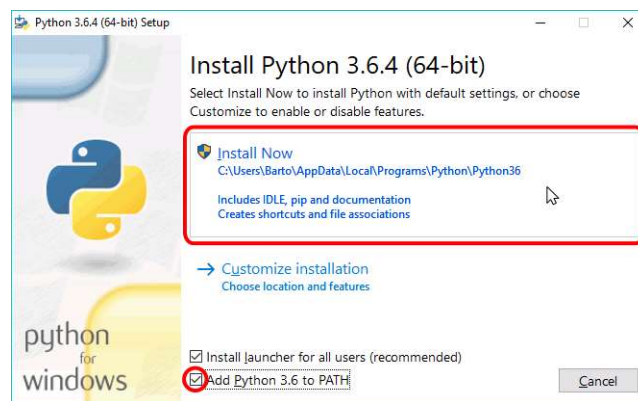
Per a varies opcions que vaig utilitzar al Python calien mòduls. Un mòdul és com una extensió del programa que et permet fer certes coses i utilitzar certes funcions que sense aquest mòdul no es podrien fer. Per a utilitzar un mòdul, al codi del Python l'has d'importar primer amb el comandament 'import'.

```
import random
```

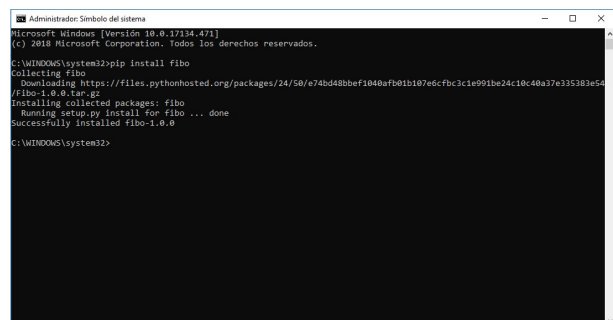
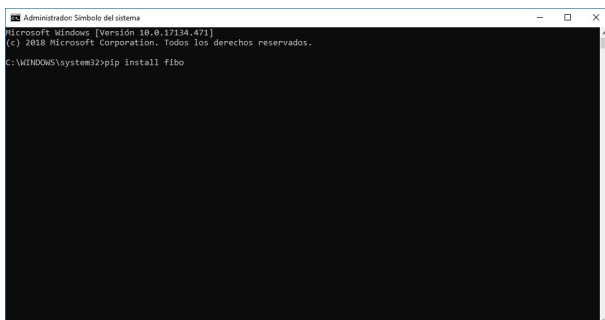
Un cop importat ja es pot utilitzar, si es tracta d'utilitzar sense importar-lo dona error.

Molts d'aquests mòduls, sobretot els més utilitzats i d'ús més genèric ja estan instal·lats per defecte. Però per a poder controlar l'Arduino des del Python em calia un mòdul anomenat serial i l'havia d'instal·lar. Per fer-ho s'havia de fer des de el CMD (símbol del sistema) executat com a administrador i utilitzant la funció pip, però no em funcionava. Vaig estar molt temps provant i no m'anava. Després d'uns dies per fi vaig trobar l'error: quan em vaig instal·lar el Python no vaig acceptar la opció que et deixa utilitzar nous comandaments al símbol del sistema que implementa Python.

Doncs, per a poder instal·lar un mòdul el primer és tenir el Python ben instal·lat. A l'hora d'instal·lar s'ha d'acceptar la opció "Añadir Python 3.6 al PATH".



Ara ja podia descarregar-me mòduls des del CMD. Havia d'utilitzar el següent comandament:

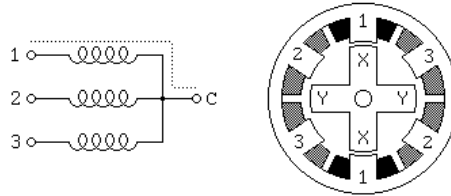


Ara ja podem utilitzar el nou mòdul!

Annex D. Funcionament dels motors pas a pas

El motor pas a pas és un motor que utilitza l'electromagnetisme per crear moviment circular. Són motors molt precisos, però no són els que tenen el major parell precisament. S'utilitzen en robòtica, impressores digitals, drons, etc.

El seu funcionament es basa en les bobines que té al voltant d'una peça en forma d'estrella. Crea un camp magnètic en dos pols i fa que s'alineïn. Al imantar el parell següent es torna a alinear girant molt poquet. Fa aquest procés successivament per crear el moviment.



Durant el treball, he utilitzat motors pas a pas monopolars i bipolars, la diferència entre els dos és principalment l'alimentació. Els motors unipolars poden tenir 5 o 6 cables de sortida, un per l'alimentació i els altres 4 per les bobines, també són més simples de controlar. El bipolar, en canvi, només en té 4 normalment.

Taula d'ordre de les fases d'un motor bipolar (sentit horari)

| Pas | A+ | A- | B+ | B- | Imatge |
|-----|----|----|----|----|--------|
| 1 | + | - | | | |
| 2 | + | - | + | - | |
| 3 | | | + | - | |
| 4 | - | + | + | - | |
| 5 | - | + | | | |
| 6 | - | + | - | + | |
| 7 | | | - | + | |
| 8 | + | - | - | + | |