

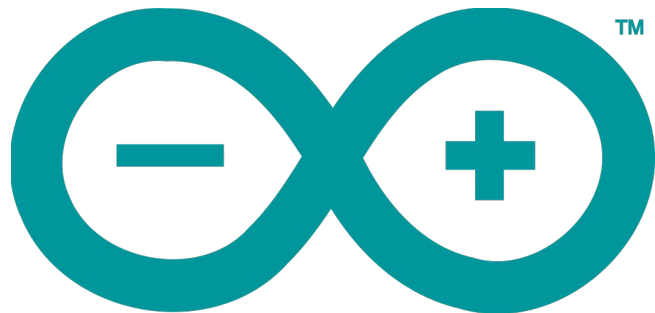


F5KFF / F6KGL

Port de Plaisance

F-93330 Neuilly sur Marne

<http://www.f6kgl-f5kff.fr>



ARDUINO

Interface Rotor K3NG



FSKFF / F6KGL

Port de Plaisance

F-93330 Neuilly sur Marne

<http://www.f6kgl-f5kff.fr>

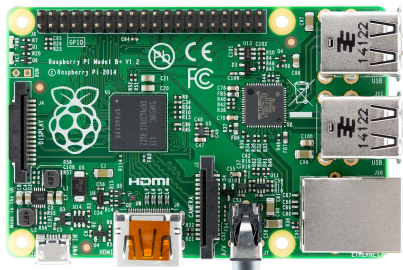
Interface rotor K3NG

- Open source
- Contrôle du rotor à distance et à moindre coût
- Large panel de fonctionnalités
- Compatibilité
- github.com/k3ng/





Comment ca marche?



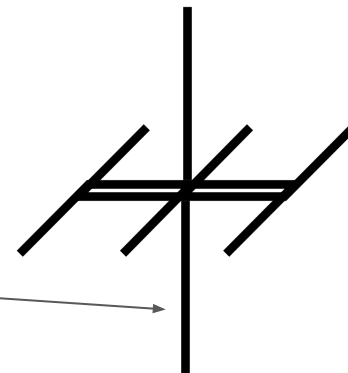
Raspberry Pi



Arduino nano



Rotor



Antenne HF



FSKFF / F6KGL

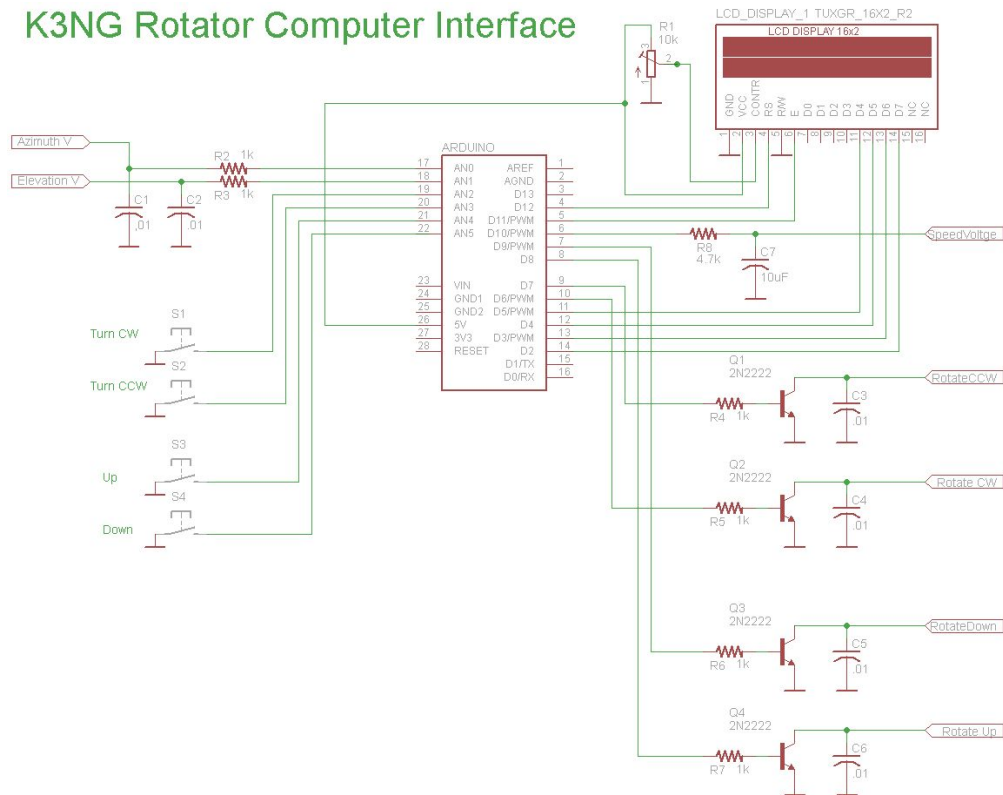
Port de Plaisance

F-93330 Neuilly sur Marine

<http://www.f6kgl-f5kff.fr>

Schéma Hardware Générique

K3NG Rotator Computer Interface





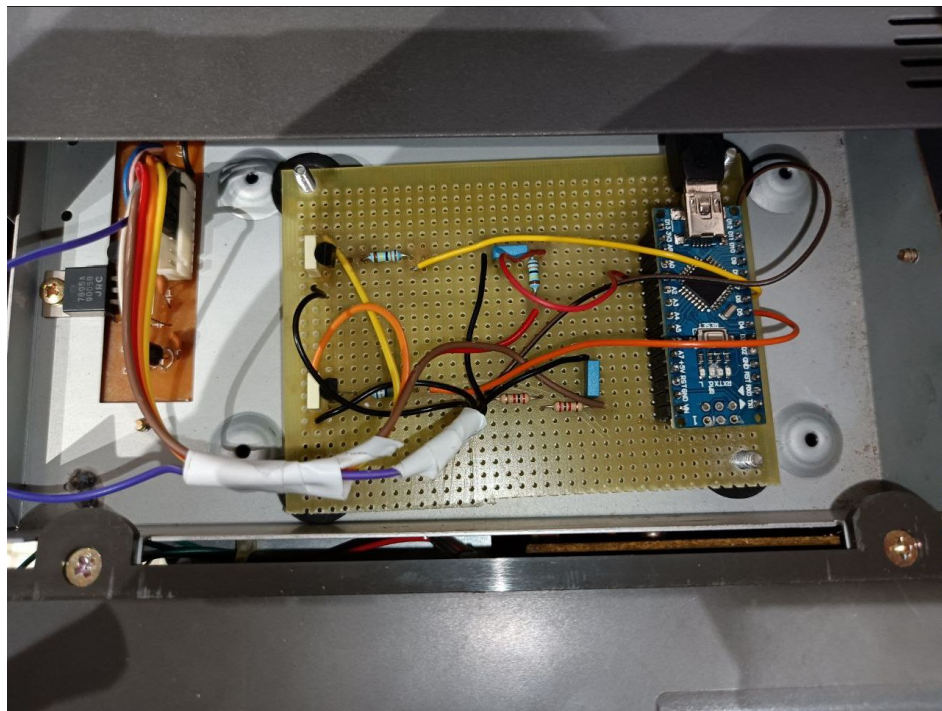
F5KFF / F6KGL

Port de Plaisance

F-93330 Neuilly sur Marine

<http://www.f6kgl-f5kff.fr>

Photos





FSKFF / F6KGL

Port de Plaisance

F-93330 Neuilly sur Seine

<http://www.f6kgl-f5kff.fr>

Configuration logicielle

Les fichiers de configuration:

rotator_features.h choisir les options de l'interface par exemple le type du protocole de communication GS-232 ou Easycom. Commenter pour désactiver ou activer une option.

```
// #define FEATURE_ELEVATION_CONTROL

#define FEATURE_YAESU_EMULATION

// #define FEATURE_EASYCOM_EMULATION

// #define FEATURE_DCU_1_EMULATION
```



FSKFF / F6KGL

Port de Plaisance

F-93330 Neuilly sur Marine

<http://www.f6kgl-f5kff.fr>

Configuration logicielle

Les fichiers de configuration:

rotator_pin.h pour définir les ports d'entrée et de sortie.

```
#define azimuth_speed_voltage 10
```

rotator_settings.h fichier de paramétrage du contrôleur.



FSKFF / F6KGL

Port de Plaisance

F-93330 Neuilly sur Seine

<http://www.f6kgl-f5kff.fr>

Configuration logicielle

Calibrage de l'interface:

- On charge une première fois le code dans l'arduino et on lance la connexion série.
- On lance le mode debug avec la commande `\d`

```
?>
\d
debug: 2021.04.07.01      154289      GS-232B
AZ:IDLE Q:- AZ:85 AZ_raw:85 Analog:118 Range:10-546 Start:15 Rotation_Capability:345 Raw_Az_Range:15-360 AZ_Speed_Norm:200 Curre
nt:200 Offset:0.00
CONFIG_NOT_DIRTY
```

- **Analog:118** Valeur du potentiomètre
- On met le rotor à 0° on récupère la valeur du potentiomètre et la même chose pour 360°



FSKFF / F6KGL

Port de Plaisance

F-93330 Neuilly sur Marne

<http://www.f6kgl-f5kff.fr>

Configuration logicielle

Calibrage de l'interface:

Dans notre cas l'antenne fait de 15° à 360°

15° -> analog 12

360° -> analog 514



FSKFF / F6KGL

Port de Plaisance

F-93330 Neuilly sur Marne

<http://www.f6kgl-f5kff.fr>

Configuration logiciel

Calibrage de l'interface:

On change les paramètre dans le fichier **rotator_settings.h**

```
#define AZIMUTH_STARTING_POINT_EEPROM_INITIALIZE 15
#define AZIMUTH_ROTATION_CAPABILITY_EEPROM_INITIALIZE 345 // 360°-15°

#define ANALOG_AZ_FULL_CCW_EEPROM_INITIALIZE 12
#define ANALOG_AZ_FULL_CW_EEPROM_INITIALIZE 514
```



FSKFF / F6KGL

Port de Plaisance

F-93330 Neuilly sur Seine

<http://www.fskff-f6kgl.fr>

Configuration logiciel

Calibrage de l'interface:

- On charge les fichier exemple -> EPROM -> Clear_eprom
- On charge les modifications et voilà !
- sur windows on peut contrôler le rotor avec PstRotorAZ
- sur Linux avec rotctl

```
rotctl -m 603 -r /dev/ttyUSB2 P 255 0
rotctl -m 603 -r /dev/ttyUSB2 p
257.00
0.00
```



FSKFF / F6KGL

Port de Plaisance

F-93330 Neuilly sur Marne

<http://www.f6kgl-f5kff.fr>

Références

- [Manuel de câblage des rotores](#)
- [GitHub Documentation](#)
- [Video Youtube k3ng Arduino rotator controller file tour](#)