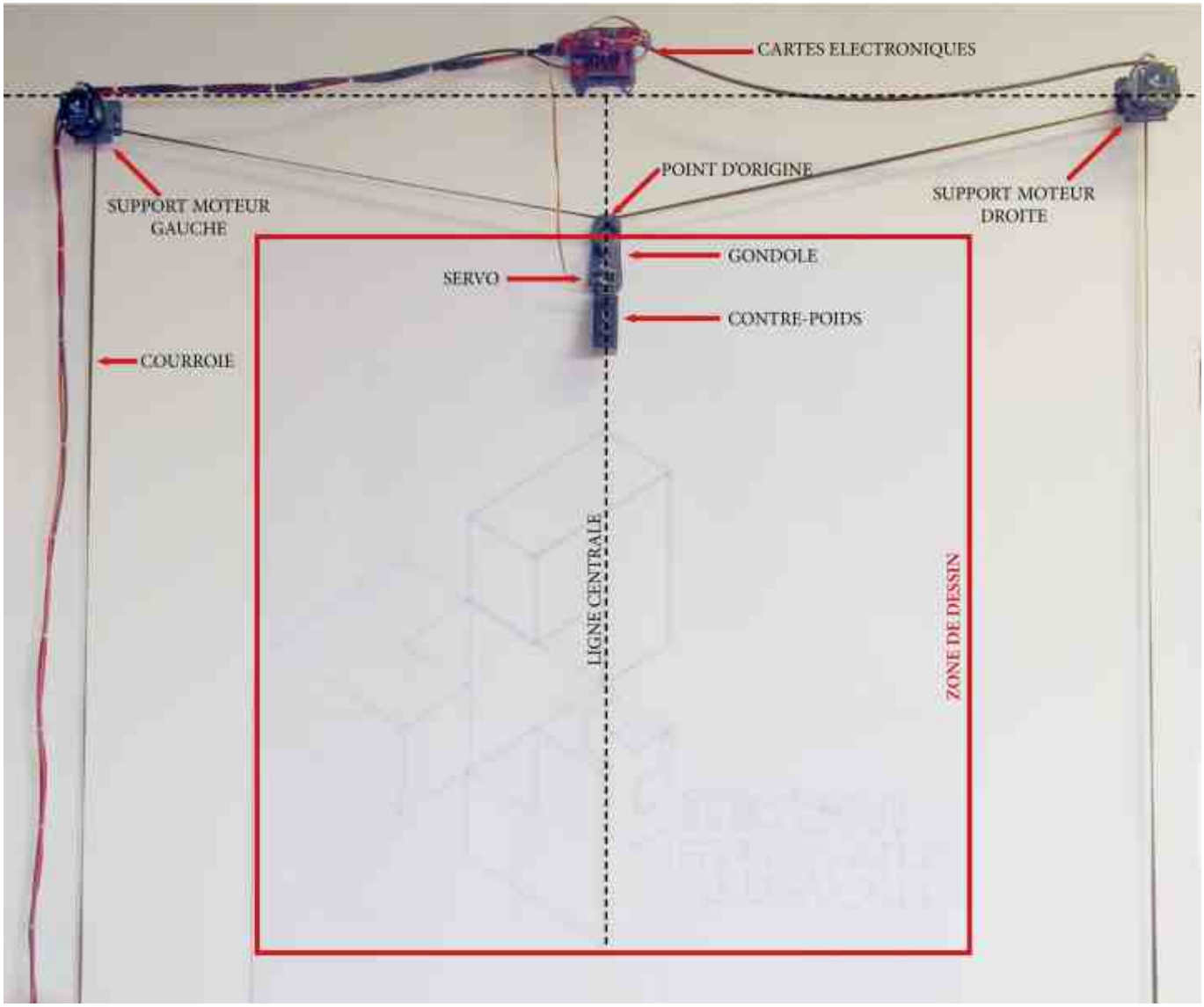
Polargraph Arduino

Wiki Polarpgraph Arduino EN

Principe



Qu'est-ce que Polargraph ?

Un Polargraph est une sorte de machine à dessiner en V suspendu vertical, ou traceur. Il est technologiquement simple, mais capable d'un travail précis à grande échelle.

Il se compose d'une plate-forme matérielle avec un microcontrôleur et quelques pilotes de moteur. Ce microcontrôleur a un logiciel écrit pour lui (appelé le micrologiciel, parfois le logiciel du serveur polargraph). Le firmware est chargé sur le microcontrôleur. Cette plate-forme matérielle est appelée "la machine polargraph".

Il existe une petite application appelée Polargraph Controller, utilisée pour envoyer des instructions à la machine. Les images elles-mêmes sont trop grandes et compliquées pour que la machine elle-même les comprenne, donc le contrôleur Polargraph les charge à la place et les décompose en

éléments d'image plus grands, ou en patchs d'ombrage, les envoyant un par un à la machine pour qu'ils les dessinent réellement. Le contrôleur Polargraph est écrit en tant qu'application de traitement, mais est disponible en tant qu'application Java précompilée sur PC, Mac et Linux.

Le site principal de Polargraph est http://www.polargraph.co.uk .

Le code source des éléments logiciels et matériels est sur github, actuellement distribué entre un certain nombre de dépôts,

Qu'est-ce qu'il y a dans un nom?

Polargraph est une marque particulière de machine à dessiner. Il est souvent utilisé de manière générique pour décrire l'un de ces types de machines, et je considère cela comme un compliment à l'omniprésence du projet, mais il ne devrait pas vraiment être utilisé, sauf pour faire référence à ce projet Polargraph.

Commandes et réponses de la machine Polargraph

Les commandes sont envoyées en série, en ascii. Ils prennent la forme :

```
<command>,[<param1>,<param2>,<param3>,<param4>,]END
```

Où les éléments entre crochets sont facultatifs.

Chaque commande se termine par un caractère ASCII 10(nouvelle ligne \nou 0x0A)

Le client doit attendre de recevoir un message READY avant d'envoyer une commande.

Sauf indication contraire, les distances sont spécifiées en pas de moteur.

Modifier la longueur :

```
C01,<left target length>,<right target length>,END
```

Déplacez les moteurs à la vitesse maximale jusqu'à ce que les cordons de la machine soient à la distance cible des moteurs, telle que mesurée en pas de moteur.

Changer la largeur de la pointe du stylet :

```
C02,<pen tip size>,END
```

réglez la pointe du stylet sur une nouvelle taille. L'unité ici est le millimètre et doit être une décimale. Il se lit comme un floatinterne.

Dessiner un pixel ombré en forme d'onde carrée :

```
C05,<left distance>,<right distance>,<size>,<brightness>,END
```

Ombrez un pixel pour qu'il ait une luminosité particulière. Un pixel est dans ce cas un patch "carré", dont les côtés sont perpendiculaires aux axes de la machine. C'est-à-dire qu'il s'agit d'un carré dessiné à l'aide du système de coordonnées natif. Il s'agit généralement d'un diamant écrasé dans l'espace cartésien.

Il est spécifié avec une position (c'est le centre), une taille (nombre de pas le long de ses côtés) et une valeur de luminosité (à quel point il est clair ou sombre). La position et la taille sont mesurées en pas de moteur. Un pixel d'onde carrée est ombré avec un motif d'onde carrée. Un pixel très sombre a de nombreuses ondes pleine largeur, un pixel très lumineux n'a pas d'ondes. La luminosité est spécifiée sous la forme d'une valeur d'échelle de gris, de 0 (noir) à 255 (blanc), où le blanc est fourni par du papier blanc et le noir par de l'encre noire.

Le mappage de la luminosité au nombre d'ondes est contrôlé dans le micrologiciel Polargraph Server et est généralement lié à la largeur du stylo pour éviter la saturation de l'encre sur les pixels très denses. :

```
C05,1000,784,55,11,END
```

// Draw a pixel 55 steps square, and with it's centre 1000 steps

// from the left-hand motor, and 784 steps from the right-hand motor.

// Given what you know about the pen installed, shade it using a

// square wave pattern so it is something like 11 255ths covered in ink.

Dessiner un pixel ombré en gribouillis :

```
C06,<left distance>,<right distance>,<size>,<brightness>,END
```

Ombragez un pixel à l'aide d'un motif de mouvement de stylet aléatoire. Cela griffonne essentiellement dans la zone des pixels pour obtenir le même type de densité d'encre que C05.

Cela utilise la largeur du stylo pour décider du nombre maximum de gribouillis exactement de la même manière que C05 l'utilise pour décider du nombre maximum de vagues. Cela a du sens avec C05 car il ne chevauche jamais les lignes - cependant, le pixel de gribouillis chevauche souvent les lignes, il est donc possible que vous puissiez griffonner toute la journée et ne jamais atteindre une couverture complète (en raison du caractère aléatoire du motif sur la même zone).

Cela rend son comportement d'ombrage un peu bizarre. Je divise généralement par deux la largeur du stylo avant de dessiner avec un pixel de gribouillis, sinon les choses sont trop claires. L'augmentation de la mise à l'échelle des pixels fera que les bords des pixels se fondent également dans ceux de leur voisin, ce qui a un bel effet de gribouillis.

Changer la direction de dessin :

```
C07,<direction mode>,<direction>,END
```

Le pixel d'onde carrée (C05) a une direction dans laquelle l'onde est dessinée, soit le long d'un axe A (le long d'un arc qui est à un rayon particulier du moteur de gauche), soit le long de l'axe B (depuis le moteur droit). Cette commande contrôle la manière dont cette direction est choisie avec le direction modeparamètre, où les options sont :

1. Automatique - tente de décider de la meilleure façon de le faire en fonction de l'endroit où il a été et de sa destination. Cela ne fonctionne pas très bien.
2. Preset - utilise la dernière direction explicitement définie.
3. Aléatoire - Compose au fur et à mesure.

Le deuxième paramètre est la direction réelle, si une doit être définie, par exemple, si direction mode est 2 (préréglé).

1. Direction Nord-Est : Le long de l'axe A, en diminuant l'axe B.
2. Direction sud-est : Le long de l'axe B, en augmentant l'axe A.
3. Direction sud-ouest : Le long de l'axe A, en augmentant l'axe B.
4. Direction nord-ouest : Le long de l'axe B, en diminuant l'axe A.

Tester la largeur du stylet à l'aide de pixels carrés :

```
C11,<pixel size>,<start pen tip size>,<end pen tip size>,<pen tip size increment>,END
```

dessine une série de pixels de test, ombrés à une densité maximale à l'aide d'un motif d'onde carrée, mais avec un paramètre de largeur de stylet variable. Cela peut être utilisé pour mesurer la taille réelle de la pointe du stylet, mais est vraiment utilisé pour calibrer votre "point noir", pour contrôler quelle est votre densité d'encre maximale. Il ne s'agit pas nécessairement d'une couche d'encre complète, ou il peut y avoir de beaux effets que vous pouvez obtenir en superposant l'encre.

La taille de pixel est la taille en pas de moteur. La taille de la pointe du stylet de départ est la taille à laquelle la pointe du stylet sera définie pour le tout premier pixel, et la taille de la pointe du stylet de fin est la taille de la pointe du stylet dans le pixel final. L' incrément de la taille de la pointe du stylet contrôle le nombre de pixels entre le début et la fin.

C'est à vous de noter à côté de chaque pixel dessiné quelle était la taille de la pointe du stylet, puis de choisir le patch que vous souhaitez comme densité maximale et de le définir dans la machine à l'aide de C02.

Définir la position du stylet :

```
C09,<left distance>,<right distance>,END
```

Indique simplement à la machine où se trouve déjà le stylet. Ceci est utilisé principalement pendant le référencement. Réglez le stylet sur une position particulière pré-mesurée, puis le réglage d'origine envoie une commande C09.

Stylet vers le bas :

```
C13,[<servo position>,]END
```

Déplace le servomoteur de levage du stylet en position basse , abaissée ou de dessin , c la pointe du stylet contre la page. La machine a une position de configuration enregistrée dans l'EEPROM, mais étant donné un paramètre entier, elle déplace le servo vers cette position à la place et utilisera cette position jusqu'à la prochaine réinitialisation de la machine - même si cela n'est pas indiqué explicitement lors des appels C13 suivants. . Utilisez C45 pour enregistrer des positions de servo de levage spécifiques dans l'EEPROM.

Stylo vers le haut :

```
C14,[<servo position>,]END
```

Déplace le servomoteur de levage du stylet en position haute , relevée ou relevée . Le paramètre se comporte de la même manière que C13.

Déplacer le stylet directement :

```
C17,<left target length>,<right target length>,<line segment lenght>,END
```

Déplace le stylet vers la cible, en traçant une ligne droite dans l'espace cartésien. Il s'agit de votre "tracer une ligne droite vers un point" de base. Les coordonnées sont toujours dans le système de coordonnées natif, pas dans les coordonnées cartésiennes. La ligne est découpée en plusieurs petites lignes, et bien que chacune ne soit pas exactement droite, dans l'ensemble, le chemin est droit. Le line segment lengthparamètre contrôle la longueur des sous-lignes. Des valeurs plus élevées signifient un dessin légèrement plus rapide, mais des lignes plus irrégulières. La valeur par défaut est une petite valeur, par exemple 2. Vous ne pouvez pas vraiment aller plus bas que cela.

Commandes de configuration de la machine

Ces paramètres sont enregistrés dans l'EEPROM interne de la machine et automatiquement rechargés lors de la réinitialisation.

Définir la taille de la machine :

```
C24,<width>,<height>,END
```

Définit les dimensions de la machine en millimètres.

Obtenez les détails actuels de la machine :

```
C26,END
```

Demandez à la machine d'envoyer quelques réponses, nous indiquant tous ses sales secrets - sa taille, etc. Ces réponses sont détaillées ci-dessous dans la section Réponses de cette page.

Réinitialiser l'EEPROM :

```
C27,END
```

Réinitialiser les paramètres d'usine par défaut.

Régler l'extension millimétrique de la machine par tour de moteur :

```
C29,<mm per rev>,END
```

Nombre entier. Définit la quantité de cordon déroulée (ou enroulée) avec un tour du moteur. C'est l'une des mesures les plus importantes à prendre correctement. C'est nominalement la circonférence des pignons d'entraînement du moteur, mais l'interaction des perles dans les dents du pignon rendra cela inexact.

Définir les pas du moteur par tour :

```
C30,<steps per rev>,END
```

Entier. La plupart des moteurs ont 200 pas par révolution, mais cela peut être multiplié avec certains types de moteurs pas à pas. S'il est possible de créer des étapes intermédiaires avec le microstepping, je recommanderais de ne pas le faire. Utilisez plutôt C37 ci-dessous pour ajouter un multiplicateur de pas. Oui, la résolution en souffrira, mais cela ne fait généralement aucune différence.

Définir la vitesse maximale du moteur :

```
C31,<max motor speed>,END
```

mesurée en pas par seconde.

Définir la vitesse d'accélération :

```
C32,<acceleration speed>,END
```

mesurée en pas par seconde, par seconde. Sonorité belle et musicale.

Définir le multiplicateur de pas :

```
 C37,<multiplier>,END
```

il s'agit d'un nombre global par lequel toutes les coordonnées entrantes sont multipliées à l'arrivée. Ceci permet à la machine de fonctionner en interne à une résolution plus élevée que celle dans laquelle elle est adressée. L'objectif principal est de permettre une utilisation pratique du micropas.

Avec le micropas, chaque pas du moteur est divisé en plusieurs pas plus petits, et il sert à se déplacer plus facilement entre les pas. C'est plus silencieux et plus doux. Ces étapes intermédiaires sont moins positives que les étapes complètes et ne sont donc pas fiables, sauf comme moyen de se déplacer entre les étapes complètes. Ce n'est pas une bonne idée de s'arrêter sur un micropas, et c'est la raison pour laquelle la machine Polargraph n'est adressable qu'en utilisant des pas de moteur complets, même si en interne elle utilise des micropas pour tous les calculs.

Polarshield utilise un micropas 8x, cette valeur doit donc être définie sur 8 pour Polarshield. Adafruit Motorshield utilise un micropas 16x, cette valeur doit donc être définie sur 16 dans ces cas.

Réponses READY

Le firmware du serveur Polargraph répondra par un READY lorsqu'il sera prêt à accepter une autre commande. Cela peut également être paramétré avec la version ou les capacités du serveur.

1. READY désigne une machine en marchepolargraph_server_a1
2. READY_100 signifie qu'il est en cours d'exécutionpolargraph_server_mega
3. READY_200 signifie qu'il est en cours d'exécutionpolargraph_server_polarshield

Ces messages sont utilisés dans le contrôleur Polargraph pour limiter les fonctionnalités disponibles.

Synchroniser

Le serveur Polargraph signale souvent la position du stylo pour permettre au contrôleur de vous renvoyer quelque chose d'utile.

SYNC,3000,4000,END

Indique que la machine estime que le stylet est à 3000 pas du moteur gauche et à 4000 pas du moteur droit.

From:
https://chanterie37.fr/fablab37110/ - **Castel'Lab le Fablab MJC de Château-Renault**

Permanent link:
https://chanterie37.fr/fablab37110/doku.php?id=start:arduino:polargraph

Last update: **2022/06/21 14:49**