

## **Manuel de l'utilisateur V2.5 17/04/2014**

# Table des matières

<b>1</b>	<b>Avant-propos</b>	<b>1</b>
<b>2</b>	<b>LinuxCNC</b>	<b>3</b>
2.1	Introduction	3
2.2	Comment fonctionne LinuxCNC	3
2.3	Interfaces utilisateur graphiques	4
2.4	Panneaux de contrôle virtuels	10
2.5	Langues	10
2.6	Penser comme un opérateur sur CNC	10
2.7	Modes opératoires	11
<b>I</b>	<b>Les interfaces utilisateur</b>	<b>12</b>
<b>3</b>	<b>L'interface graphique AXIS</b>	<b>13</b>
3.1	Introduction	13
3.2	Commencer avec AXIS	14
3.2.1	Une session typique avec AXIS	14
3.3	Eléments de la fenêtre d'AXIS	15
3.3.1	Options des menus	15
3.3.1.1	Menu Fichier	15
3.3.1.2	Menu Machine	16
3.3.1.3	Menu Vues	17
3.3.1.4	Menu Aide	19
3.3.2	Boutons de la barre d'outils	19
3.3.3	Zones d'affichage graphique du programme	20
3.3.3.1	Affichage des coordonnées	20
3.3.3.2	Vue du parcours d'outil	20
3.3.3.3	Étendues du programme	20
3.3.3.4	Le cône d'outil	21
3.3.3.5	Parcours d'outil	21

---

---

3.3.3.6	Interaction avec l'affichage	21
3.3.4	Zone texte d'affichage du programme	21
3.3.5	Contrôle manuel	22
3.3.5.1	Le groupe de cases et boutons <i>Axes</i>	22
3.3.5.2	La prise d'origine machine	23
3.3.5.3	Toucher	23
3.3.5.4	Dépassement de limite	23
3.3.5.5	Le groupe <i>Broche</i>	24
3.3.5.6	Le groupe <i>Arrosage</i>	24
3.3.6	Données manuelles (MDI)	24
3.3.7	Correcteurs de vitesse	25
3.3.8	Correcteur de vitesse de broche	25
3.3.9	Vitesse de Jog	25
3.3.10	Vitesse Maxi	25
3.4	Raccourcis clavier	25
3.5	Affichage de l'état de LinuxCNC (LinuxCNCtop)	26
3.6	Entrée de données en texte (MDI)	27
3.7	axis-remote: Commande à distance de l'interface graphique d'AXIS	28
3.8	hal_manualtoolchange: Dialogue de changement d'outil manuel	28
3.9	Modules en Python	28
3.10	Utiliser AXIS en mode tour	28
3.11	Configuration avancée d'AXIS	29
3.11.1	Filtres de programme	29
3.11.2	La base de données des ressources X	30
3.11.3	Manivelle de jog	31
3.11.4	~/.axisrc	31
3.11.5	Éditeur externe	31
3.11.6	Panneau de contrôle virtuel	31
3.11.7	Commentaires spéciaux	31
<b>4</b>	<b>L'utilitaire graphique NGCGUI</b>	<b>33</b>
4.1	Vue d'ensemble	33
4.2	Configurations fournies en exemple.	34
4.3	Librairies	34
4.4	Intégration de ngcgui dans Axis	35
4.4.1	Traceur Truetype	36
4.4.2	Exemples d'INI	37
4.5	Besoins des sous-programmes	39
4.6	Exemple, découpe pour DB25	41
4.7	Création d'un sous-programme	44

---

---

<b>5</b>	<b>L'interface graphique Touchy</b>	<b>45</b>
5.1	Panneau de Configuration . . . . .	45
5.1.1	Connections avec HAL . . . . .	45
5.1.1.1	Contrôles requis . . . . .	46
5.1.1.2	Contrôles optionnels . . . . .	46
5.1.1.3	Voyants de panneau optionnels . . . . .	46
5.1.2	Recommandé pour toutes les configurations . . . . .	46
5.2	Réglages . . . . .	46
5.2.1	Macros . . . . .	46
<b>6</b>	<b>L'interface graphique TkLinuxCNC</b>	<b>48</b>
6.1	Introduction . . . . .	48
6.2	Utiliser TkLinuxCNC . . . . .	49
6.2.1	Une session typique avec TkLinuxCNC . . . . .	49
6.3	Éléments affichés par TkLinuxCNC . . . . .	50
6.3.1	Boutons principaux . . . . .	50
6.3.2	Barre de statut des différents offsets . . . . .	50
6.3.3	Zone d'affichage des coordonnées . . . . .	50
6.3.3.1	Parcours d'outil . . . . .	51
6.3.4	Contrôle en automatique . . . . .	51
6.3.4.1	Boutons de contrôle . . . . .	51
6.3.4.2	Zone texte d'affichage du programme . . . . .	51
6.3.5	Contrôle en manuel . . . . .	51
6.3.5.1	Touches implicites . . . . .	51
6.3.5.2	Le groupe de boutons <i>Broche</i> . . . . .	52
6.3.5.3	Le groupe de boutons <i>Arrosage</i> . . . . .	52
6.3.6	Entrée manuelle de G-code (MDI) . . . . .	52
6.3.6.1	MDI: . . . . .	53
6.3.6.2	G-Codes actifs . . . . .	53
6.3.7	Vitesse de Jog . . . . .	53
6.3.8	Correcteur de vitesse d'avance travail . . . . .	53
6.3.9	Correcteur de vitesse de broche . . . . .	53
6.4	Raccourcis clavier . . . . .	53
<b>7</b>	<b>L'interface graphique MINI</b>	<b>55</b>
7.1	Introduction . . . . .	55
7.2	L'écran . . . . .	56
7.3	Barre de menu . . . . .	56
7.4	Barre de boutons de contrôle . . . . .	58

---

---

7.4.1	MANUEL	58
7.4.2	AUTO	59
7.4.3	MDI	60
7.4.4	VITESSE-0 et CONTINUER	60
7.4.5	ABANDON	60
7.4.6	Arrêt d'Urgence	60
7.5	Colonne de gauche	61
7.5.1	Afficheurs de position des axes	61
7.5.2	Correcteur de vitesse travail	61
7.5.3	Messages	61
7.6	Colonne de droite	61
7.6.1	Editeur de texte	62
7.6.2	Parcours d'outil	63
7.6.3	Page des outils	64
7.6.4	Page des offsets	64
7.7	Raccourcis clavier	65
7.7.1	Touches courantes	65
7.7.2	Mode Manuel	66
7.7.3	Mode Auto	67
7.8	Divers	67
<b>8</b>	<b>L'interface textuelle KEYSTICK</b>	<b>68</b>
8.1	Introduction	68
8.2	Installation	69
8.3	Utilisation	69
<b>II</b>	<b>L'utilisation de LinuxCNC</b>	<b>70</b>
<b>9</b>	<b>Concepts importants pour l'utilisateur</b>	<b>71</b>
9.1	La configuration machine	71
9.2	Contrôle de trajectoire	73
9.2.1	La planification de trajectoire	73
9.2.2	Le suivi du parcours	74
9.2.3	La programmation du planificateur	74
9.2.4	Planification des mouvements	75
9.3	G-code	75
9.3.1	Par défaut	75
9.4	Vitesse d'avance	75
9.5	Compensation de rayon d'outil	76

---

---

9.6	Prise d'origine machine	76
9.7	Changement d'outil	76
9.8	Systèmes de coordonnées	76
9.8.1	G53 Coordonnées machine	76
9.8.2	G54 à 59.3 Coordonnées utilisateur	76
9.8.3	Quand vous êtes perdu	77
<b>10</b>	<b>Aperçu global d'une machine CNC</b>	<b>78</b>
10.1	Composants mécaniques	78
10.1.1	Axes	78
10.1.1.1	Axes linéaires primaires	78
10.1.1.2	Axes linéaires secondaires	78
10.1.1.3	Axes rotatifs	78
10.1.2	Broche	79
10.1.3	Arrosages	79
10.1.4	Correcteurs de vitesse d'avance et de broche	79
10.1.5	Bouton d'effacement de bloc	79
10.1.6	Bouton d'arrêt optionnel du programme	79
10.2	Composants de contrôle et de données	79
10.2.1	Axes linéaires	79
10.2.2	Axes linéaires secondaires	79
10.2.3	Axes rotatifs	79
10.2.4	Point contrôlé	79
10.2.5	Mouvement linéaire coordonné	80
10.2.6	Vitesse d'avance	80
10.2.7	Arrosage	80
10.2.8	Temporisation	80
10.2.9	Unités	80
10.2.10	Position courante	80
10.2.11	Choix du plan de travail	81
10.2.12	carrousel d'outils	81
10.2.13	Changeur d'outil	81
10.2.14	Chargeur de pièce	81
10.2.15	Chargeur de pièces	81
10.2.16	Boutons des correcteurs de vitesses	81
10.2.17	Modes de contrôle de trajectoire	81
10.3	Interaction de l'interpréteur avec les boutons	81
10.3.1	Boutons de correction de vitesses	81
10.3.2	Bouton d'effacement de bloc	82
10.3.3	Bouton d'arrêt optionnel du programme	82

---

<b>11 Systèmes de coordonnées et décalages</b>	<b>84</b>
11.1 Introduction	84
11.2 Commande en coordonnées machine: G53	84
11.3 Décalages pièce (G54 à G59.3)	85
11.3.1 Système de coordonnées par défaut	86
11.3.2 Réglage des décalages avec G10	86
11.4 Décalages d'axes G92	86
11.4.1 Les commandes G92	87
11.4.2 Réglage des valeurs de G92	87
11.4.3 Précautions avec G92	87
11.5 Exemple de programme utilisant les décalages d'axes	88
<b>12 Les compensations d'outil</b>	<b>90</b>
12.1 Compensation de longueur d'outil	90
12.1.1 Toucher	90
12.1.2 Utilisation de G10 L1/L10/L11	91
12.2 Table d'outils	91
12.2.1 Format de la table d'outils	91
<b>13 Fichier d'outils et compensations</b>	<b>93</b>
13.1 Fichier d'outils	93
13.2 Compensation d'outil	93
13.3 Compensation de longueur d'outil	93
13.4 Compensation de rayon d'outil	94
13.4.1 Vue générale	95
13.4.1.1 Table d'outils	95
13.4.1.2 Programmation des mouvements d'entrée	95
13.4.1.3 Mouvement en Z	96
13.4.1.4 Mouvement en vitesse rapide	96
13.4.1.5 Bonne pratique	96
13.4.2 Exemples de profils	97
13.4.2.1 Profil extérieur	97
13.4.2.2 Profil intérieur	98
13.5 Deux exposés sur les compensations d'outil	98
13.5.1 Compensation de rayon d'outil, détails	98
13.5.1.1 Instructions de programmation	99
13.5.1.2 La valeur de D	99
13.5.1.3 Table d'outils	100
13.5.1.4 Deux types de contour	100

13.5.1.5	Contour sur le profil du matériau	100
13.5.1.6	Contour sur le parcours d'outil	100
13.5.1.7	Erreurs de programmation et limitations	101
13.5.2	Premier mouvement	102
13.5.2.1	Programmation des mouvements d'entrée	103
13.5.2.2	Méthode générale	103
13.5.2.3	Méthode simple	104
13.5.2.4	Autres points où est exécutée la compensation de rayon d'outil	105
13.5.2.5	Algorithmes pour compensation de rayon d'outil	105
13.5.3	Exemples de Jon Elson	105
<b>14</b>	<b>Vue générale du langage G-codes de LinuxCNC</b>	<b>108</b>
14.1	Brève description du G-code de LinuxCNC	108
14.2	Format des paramètres du G-code	108
14.3	Format d'une ligne	108
14.4	Caractère d'effacement de bloc	109
14.5	Numéro de ligne	109
14.6	Les mots	109
14.7	Les nombres	110
14.8	Paramètres (Variables)	110
14.9	Paramètres numérotés	111
14.10	Paramètres de sous-programme	112
14.11	Paramètres nommés	112
14.12	Paramètres système	113
14.13	Expressions	113
14.14	Opérateurs binaires	113
14.15	Fonctions	113
14.16	Répétitions d'items	114
14.17	Ordre des items	114
14.18	Commandes et modes machine	115
14.19	Coordonnées polaires	115
14.20	Groupes modaux	117
14.21	Commentaires	118
14.22	Messages	119
14.23	Enregistrement des mesures	119
14.24	Log général	119
14.25	Messages de débogage	119
14.26	Paramètres dans les commentaires	119
14.27	Exigences des fichiers	120

14.28	Taille des fichiers	120
14.29	Ordre d'exécution	120
14.30	G-Code: Bonnes pratiques	121
14.30.1	Utiliser un nombre de décimales approprié	121
14.30.2	Utiliser les espaces de façon cohérente	121
14.30.3	Préférer le <i>format centre</i> pour les arcs	121
14.30.4	Placer les codes modaux importants au début des programmes	121
14.30.5	Ne pas mettre trop de choses sur une ligne	121
14.30.6	Ne pas régler et utiliser un paramètre sur la même ligne	121
14.30.7	Ne pas utiliser les numéros de ligne	121
14.30.8	Lorsque plusieurs systèmes de coordonnées sont déplacés	121
14.31	Axes rotatifs et linéaires	122
14.32	Messages d'erreur courants	122

## **15 Tout le G-code de LinuxCNC** **123**

15.1	Conventions d'écriture du G-code	123
15.2	Table d'index du G-code	123
15.3	G0 Interpolation linéaire en vitesse rapide	124
15.4	G1 Interpolation linéaire en vitesse travail	125
15.5	G2, G3 Interpolation circulaire en vitesse travail	125
15.5.1	Arc au format centre (format recommandé)	126
15.5.2	Exemples d'arcs au format centre	127
15.5.3	Arcs au format rayon (format non recommandé)	129
15.6	G4 Tempo	130
15.7	G5.1 B-spline quadratique	130
15.8	G5.2 G5.3 NURBs Block	130
15.9	G7 Mode diamètre sur les tours	132
15.10	G8 Mode rayon sur les tours	132
15.11	G10 L1 Ajustements dans la table d'outils	132
15.12	G10 L2 Établissement de l'origine d'un système de coordonnées	132
15.13	G10 L10 modifie les offsets d'outil dans la table d'outils	133
15.14	G10 L11 modifie les offsets d'outil dans la table d'outils	134
15.15	G10 L20 Établissement de l'origine d'un système de coordonnées	134
15.16	G17 à G19.1 Choix du plan de travail	135
15.17	G20, G21 Choix des unités machine	135
15.18	G28, G28.1 Aller à une position prédéfinie	135
15.19	G30, G30.1 Aller à une position prédéfinie	135
15.20	G33 Mouvement avec broche synchronisée	136
15.21	G33.1 Taraudage Rigide	137

15.22G38.x Mesure au palpeur . . . . .	138
15.23G40 Révocation de la compensation de rayon d'outil . . . . .	139
15.24G41, G42 Compensation de rayon d'outil . . . . .	139
15.25G41.1, G42.1 Compensation dynamique d'outil . . . . .	139
15.26G43 Activation de la compensation de longueur d'outil . . . . .	140
15.27G43.1 Compensation dynamique de longueur d'outil . . . . .	140
15.28G49 Révocation de la compensation de longueur d'outil . . . . .	140
15.29G53 Mouvement en coordonnées absolues . . . . .	141
15.30G54 à G59.3 Choix du système de coordonnées . . . . .	141
15.31G61, G61.1 Contrôle de trajectoire exacte . . . . .	142
15.32G64 Contrôle de trajectoire continue avec tolérance . . . . .	142
15.33G73 Cycle de perçage avec brise copeaux . . . . .	142
15.34G76 Cycle de filetage préprogrammé . . . . .	143
15.35Les cycles de perçage G81 à G89 . . . . .	145
15.35.1 Mots communs . . . . .	145
15.35.2 Mots <i>sticky</i> . . . . .	145
15.35.3 Répétition de cycle . . . . .	146
15.35.4 Mode de retrait . . . . .	146
15.35.5 Erreurs des cycles de perçage . . . . .	146
15.35.6 Mouvement préliminaire et Intermédiaire . . . . .	146
15.35.7 Pourquoi utiliser les cycles de perçage? . . . . .	146
15.36G80 Révocation des codes modaux . . . . .	147
15.37G81 Cycle de perçage . . . . .	148
15.38G82 Cycle de perçage avec temporisation . . . . .	151
15.39G83 Cycle de perçage avec déburrage . . . . .	151
15.40G84 Cycle de taraudage à droite . . . . .	152
15.41G85 Cycle d'alésage, sans temporisation, retrait en vitesse travail . . . . .	152
15.42G86 Cycle d'alésage, arrêt de broche, retrait en vitesse rapide . . . . .	152
15.43G87 Alésage inverse . . . . .	152
15.44G88 Alésage, arrêt de broche, retrait en manuel . . . . .	152
15.45G89 Cycle d'alésage, temporisation, retrait en vitesse travail . . . . .	153
15.45.1 Pourquoi utiliser les cycles de perçage ? . . . . .	153
15.46G90, G91: Modes de déplacement . . . . .	154
15.47G90.1, G91.1: Mode de déplacement en arc (I, J et K) . . . . .	154
15.48G92 Décalage d'origine des systèmes de coordonnées . . . . .	155
15.49G92.1, G92.2 Remise à zéro des décalages des systèmes de coordonnées . . . . .	155
15.50G92.3 Restauration des décalages d'axe . . . . .	155
15.51G93, G94, G95: Choix des modes de vitesse . . . . .	155
15.52G96, G97: Modes de contrôle de la broche . . . . .	156
15.53G98, G99: Options du plan de retrait . . . . .	156

---

<b>16 Les M-codes</b>	<b>158</b>
16.1 Table des M-codes	158
16.2 M0, M1, pause dans le programme	158
16.3 M2, M30, fin de programme	159
16.4 M60, pause pour déchargement pièce	159
16.5 M3, M4, M5 Contrôle de la broche	159
16.6 M6 Appel d'outil	159
16.6.1 Changement d'outil manuel	159
16.6.2 Changement d'outil	160
16.7 M7, M8, M9 Contrôle de l'arrosage	160
16.8 M48, M49 Contrôle des correcteurs de vitesse	160
16.9 M50 Contrôle du correcteur de vitesse travail	160
16.10 M51 Contrôle du correcteur de vitesse broche	161
16.11 M52 Contrôle de vitesse adaptative	161
16.12 M53 Contrôle de la coupure de vitesse	161
16.13 M61 Correction du numéro de l'outil courant	161
16.14 M62 à M65 Contrôle de bits de sortie numérique	161
16.15 M66 Contrôle d'entrée numérique et analogique	162
16.16 M67 Contrôle de sortie analogique	162
16.17 M68 Contrôle de sortie analogique directe	163
16.18 M100 à M199 Commandes définies par l'utilisateur	163
<b>17 Les O-codes</b>	<b>165</b>
17.1 Utilisation des O-codes	165
17.2 Sous-programmes: <b>sub, endsub, return, call</b>	165
17.3 Boucles: <b>do, while, endwhile, break, continue</b>	166
17.4 Conditionnel: <b>if, elseif, else, endif</b>	167
17.5 Répétition: <b>Repeat</b>	167
17.6 Indirection	168
17.7 Appel de fichier	168
<b>18 Les autres codes</b>	<b>169</b>
18.1 F: Réglage de la vitesse d'avance travail	169
18.2 S: Réglage de la vitesse de rotation de la broche	169
18.3 T: Choix de l'outil	169

---

<b>19 Exemples de fichiers G-Code</b>	<b>171</b>
19.1 Exemples pour une fraiseuse	171
19.1.1 Fraisage hélicoïdal d'un orifice	171
19.1.2 Rainurage	171
19.1.3 Palpage d'une grille rectangulaire de points	172
19.1.4 Amélioration du palpage d'une grille rectangulaire de points	173
19.1.5 Mesure de longueur d'outil	174
19.1.6 Mesure d'un alésage au palpeur	174
19.1.7 Compensation de rayon d'outil	174
19.2 Exemples pour un tour	175
19.2.1 Filetage	175
<b>20 Particularités des tours</b>	<b>176</b>
20.1 Mode tour	176
20.2 Fichier d'outils	176
20.3 Orientations des outils de tournage	176
20.4 Correction d'outil	180
20.4.1 Offset d'outil en X	180
20.4.2 Séquence typique de correction d'outil en X:	180
20.4.3 Offset d'outil en Z	180
20.4.4 Séquence typique de correction d'outil en Z:	180
20.4.5 Machine avec tous les outils compensés	181
20.4.6 Séquence typique de décalage du système de coordonnées:	181
20.5 Mouvements avec broche synchronisée	181
20.6 Arcs	181
20.6.1 Les arcs et la cinématique du tour	181
20.6.2 Mode rayon et mode diamètre	182
20.7 Parcours d'outil	182
20.7.1 Point contrôlé	182
20.7.2 Tourner les angles sans compensation d'outil	182
20.7.3 Tournage avec rayon extérieur	184
20.7.4 Utiliser la compensation d'outil	186
<b>21 Différences avec RS274/NGC</b>	<b>188</b>
21.1 Changements entre RS274/NGC et LinuxCNC	188
21.1.1 Position après un changement d'outil	188
21.1.2 Les paramètres de décalage sont dans l'unité du fichier ini	188
21.1.3 Table d'outils, longueur et diamètre sont dans l'unité du fichier ini	188
21.1.4 G84, G87 ne sont pas implémentés	188

21.1.5	G28, G30 avec des mots d'axe	188
21.2	Ajouts à RS274/NGC	189
21.2.1	Codes de filetage G33 et G76	189
21.2.2	G38.2	189
21.2.3	G38.3 à G38.5	189
21.2.4	Les O-codes	189
21.2.5	M50 à M53 Correcteurs de vitesse	189
21.2.6	M61 à M66	189
21.2.7	G43, G43.1	189
21.2.8	G41.1, G42.1 Compensation dynamique	189
21.2.9	G43 sans le mot H	189
21.2.10	U, V et W axes	190
<b>22</b>	<b>Image-to-gcode: Usiner un depth maps</b>	<b>191</b>
22.1	Qu'est-ce qu'un <i>depth map</i> ?	191
22.2	Intégrer image-to-gcode dans l'interface utilisateur d'AXIS	191
22.3	Utilisation d'image-to-gcode	192
22.4	Les différentes options	192
22.4.1	Unités	192
22.4.2	Invert Image	192
22.4.3	Normalize Image	192
22.4.4	Expand Image Border	192
22.4.5	Tolerance (unités)	192
22.4.6	Pixel Size (unités)	192
22.4.7	Plunge Feed Rate (unités par minute)	192
22.4.8	Feed Rate (unités par minute)	193
22.4.9	Spindle Speed (RPM)	193
22.4.10	Scan Pattern	193
22.4.11	Scan Direction	193
22.4.12	Depth (unités)	193
22.4.13	Step Over (pixels)	193
22.4.14	Tool Diameter	193
22.4.15	Safety Height	193
22.4.16	Tool Type	193
22.4.17	Lace bounding	194
22.4.18	Contact angle	194
22.4.19	Offset d'ébauche et profondeur par passe d'ébauche	194
<b>23</b>	<b>Glossary</b>	<b>195</b>

---

<b>24 Legal Section</b>	<b>200</b>
24.1 Copyright Terms . . . . .	200
24.2 GNU Free Documentation License . . . . .	200
<b>25 Index</b>	<b>204</b>

---



### The LinuxCNC Team

Ce manuel est en évolution permanente. Si vous voulez nous aider à son écriture, sa rédaction, sa traduction ou la préparation des graphiques, merci de contactez n'importe quel membre de l'équipe de traduction ou envoyez un courrier électronique à [emc-users@lists.sourceforge.net](mailto:emc-users@lists.sourceforge.net).

Copyright © 2000–2012 LinuxCNC.org

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".. If you do not find the license you may order a copy from Free Software Foundation, Inc. 59 Temple Place, Suite 330 Boston, MA 02111-1307

LINUX® is the registered trademark of Linus Torvalds in the U.S. and other countries. The registered trademark Linux® is used pursuant to a sublicense from LMI, the exclusive licensee of Linus Torvalds, owner of the mark on a world-wide basis.

Permission est donnée de copier, distribuer et/ou modifier ce document selon les termes de la « GNU Free Documentation License », Version 1.3 ou toute version ultérieure publiée par la « Free Software Foundation »; sans sections inaltérables, sans texte de couverture ni quatrième de couverture. Une copie de la licence est incluse dans la section intitulée « GNU Free Documentation License ». Si vous ne trouvez pas la licence vous pouvez en commander un exemplaire chez Free Software Foundation, Inc. 59 Temple Place, Suite 330 Boston, MA 02111-1307

(La version de langue anglaise fait foi)

---

#### AVIS

La version Française de la documentation de LinuxCNC est toujours en retard sur l'originale faute de disponibilité des traducteurs.

Il est recommandé d'utiliser la documentation en Anglais chaque fois que possible.

Si vous souhaitez être un traducteur bénévole pour la documentation française de LinuxCNC, merci de nous contactez.

---

---

#### NOTICE

The French version of the LinuxCNC documentation is always behind the original fault availability of translators.

It's recommended to use the English documentation whenever possible.

If you would like to be a volunteer editor for the French translation of LinuxCNC, please contact us.

---

# Chapitre 1

## Avant-propos

LinuxCNC est souple et modulaire. Ces attributs l'ont fait apparaître à certains comme un brouillon de petits morceaux confus, ils se sont demandé pourquoi il en était ainsi. Cette page tente de répondre à cette question avant que vous lecteurs, ne plongiez dedans pour vous faire votre propre idée.

EMC a débuté à l'institut national des standards et des technologies des Etats Unis, le NIST. Il a mûri comme un logiciel fonctionnant sur le système d'exploitation Unix. Unix le rendait différent. Très tôt des développeurs Unix ont apporté une série d'idées concernant l'écriture du code, c'est devenu une écriture selon «la tradition d'Unix». Les premiers auteurs de LinuxCNC ont suivi cette voie.

Eric S. Raymond, dans son livre *The Art of Unix Programming*, résume la philosophie Unix par la philosophie largement utilisée en ingénierie, le principe KISS Keep it Simple, Stupid Reste Simple, Crétin ou Sois Simple et Concis. Puis il décrit sa vision selon laquelle cette philosophie globale s'applique en tant que norme culturelle Unix, bien qu'on trouve sans surprise de graves entorses à la plupart des règles Unix suivantes:

- Règle de modularité: Ecrire des éléments simples reliés par de bonnes interfaces.
- Règle de clarté: La Clarté vaut mieux que l'ingéniosité.
- Règle de composition: Concevoir des programmes qui peuvent être reliés à d'autres programmes.
- Règle de séparation: Séparer les règles du fonctionnement; Séparer les interfaces du mécanisme.<sup>1</sup>

Monsieur Raymond offre d'autres règles mais ces quatre là décrivent les caractéristiques essentielles du système de contrôle de mouvement LinuxCNC.

La règle de *Modularité* est critique. Tout au long de ces manuels, vous trouverez des discussions à propos de l'interpréteur ou à propos des planificateurs de tâche ou de mouvement ou encore à propos de HAL. Chacun d'eux est un module ou un ensemble de modules. Cette modularité vous permettra de ne connecter entre elles que les parties dont vous avez besoin pour le bon fonctionnement de votre machine.

La règle de *clarté* est essentielle. LinuxCNC est en perpétuelle évolution, il n'est pas terminé et ne le sera jamais. Il est assez complet pour piloter toutes les machines que nous avons voulu qu'il pilote. Une bonne partie de cette évolution est atteinte parce que les utilisateurs et les développeurs peuvent voir le travail des autres et construire sur ce qui est déjà fait.

La règle de *composition* nous permet de concevoir et de construire un contrôleur à partir des nombreux modules existants, en les rendant connectables entre eux. Nous obtenons cette connectivité en appliquant une interface standard à tous les modules et en suivant ce standard.

La règle de *séparation* exige que chaque petite chose soit faite par une partie distincte. En séparant les fonctions, le dépannage est rendu plus aisé, le remplacement de modules par d'autres peut être fait à l'intérieur du système et la comparaison s'effectuer facilement.

Qu'apporte la fameuse «tradition d'Unix» à vous, utilisateurs de LinuxCNC. Elle signifie que vous pourrez faire des choix sur la façon d'utiliser le système. Beaucoup de ces choix affecteront les parties intégrées à la machine, mais beaucoup également affecteront la manière dont vous utiliserez votre machine. Au cours de votre lecture, vous trouverez différents endroits où vous pourrez faire des comparaisons. Finalement vous pourrez dire «J'utiliserai cette interface plutôt que telle autre» ou, «J'écirai cette nouvelle partie de telle manière plutôt que de telle autre.» Tout au long de ces manuels nous décrirons l'étendue des possibilités de LinuxCNC actuellement disponibles.

---

1. Trouvé sur [http://fr.wikipedia.org/wiki/Philosophie\\_d%27Unix](http://fr.wikipedia.org/wiki/Philosophie_d%27Unix), 09/09/2008

Puisque vous commencez votre voyage dans l'utilisation de LinuxCNC nous vous proposons ces deux citations de précaution.

- Pour paraphraser les paroles de Doug Gwyn sur UNIX: "LinuxCNC n'a pas été conçu pour empêcher ses utilisateurs de commettre des actes stupides, car cela les empêcherait aussi de réaliser des actes ingénieux."
- De même les paroles de Steven King: "LinuxCNC est convivial. Cependant Unix ne précise pas vraiment avec qui."

## Chapitre 2

# LinuxCNC

### 2.1 Introduction

Ce document est centré sur l'utilisation de LinuxCNC, il est plutôt destiné aux lecteurs l'ayant déjà installé et configuré. Quelques informations sur l'installation sont données dans les chapitres suivants. La documentation complète sur l'installation et la configuration se trouve dans le manuel de l'intégrateur.

### 2.2 Comment fonctionne LinuxCNC

LinuxCNC est un peu plus que juste un autre programme de fraiseuse CNC. Il est capable de contrôler des machines-outils, des robots ou d'autres automatismes. Il est capable de contrôler des servomoteurs, des moteurs pas à pas, des relais ainsi que d'autres mécanismes relatifs aux machines-outils.

Il y a quatre principales composantes du logiciel LinuxCNC:

- un contrôleur de mouvement (EMCMOT),
- un contrôleur d'entrées/sorties discrètes (EMCIO),
- un exécuteur des tâches qui les coordonne (EMCTASK),
- et les interfaces utilisateur graphiques.

En outre il existe une couche appelée HAL (couche d'abstraction du matériel) qui permet la configuration de LinuxCNC sans avoir besoin de recompiler.

---

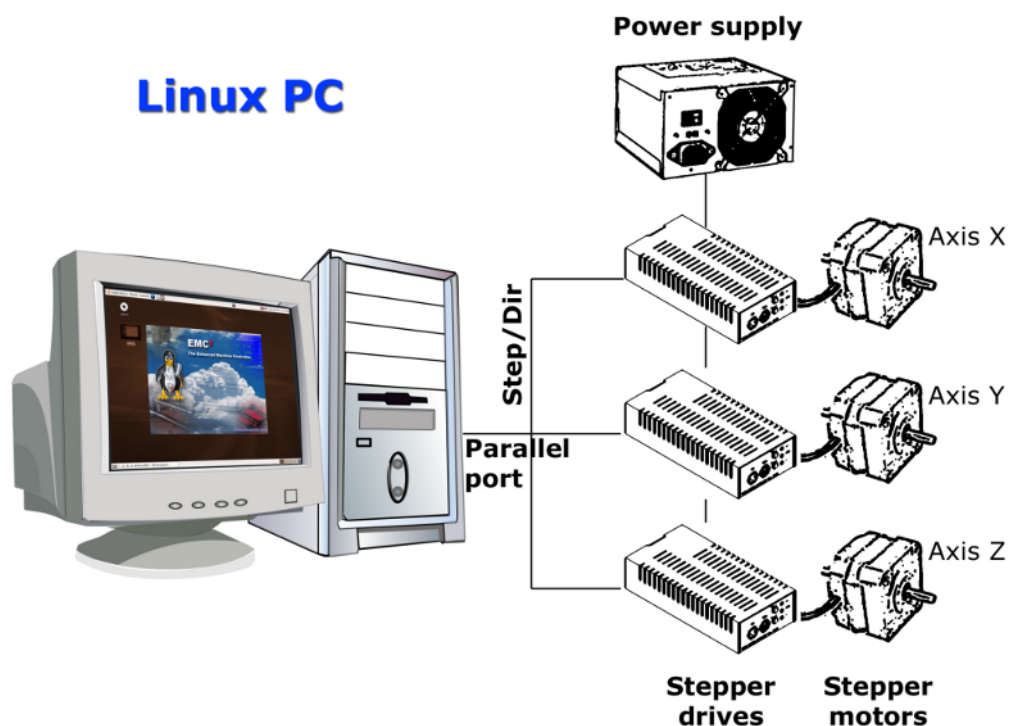


FIGURE 2.1 – Machine simple contrôlée par LinuxCNC

La figure ci-dessus montre un diagramme bloc représentant une machine 3 axes typique comme LinuxCNC les aime. Cette figure montre un système basé sur des moteurs pas à pas. Le PC, tournant sous Linux contrôle les interfaces de puissance des moteurs pas à pas en leur envoyant des signaux au travers du port parallèle. Ces signaux (impulsionnels) font que la puissance adéquate est fournie aux moteurs. LinuxCNC peut également contrôler des servomoteurs via une interface de puissance pour servomoteurs ou utiliser le port parallèle étendu connecté à une carte de contrôle externe. Quand nous examinerons chacun des composants qui forment un système LinuxCNC, nous nous référerons à cette machine typique.

## 2.3 Interfaces utilisateur graphiques

L'interface graphique est la partie de LinuxCNC qui interagit avec l'opérateur de la machine. LinuxCNC est fourni avec plusieurs interfaces utilisateurs graphiques:

- [Axis](#), l'interface utilisateur standard.

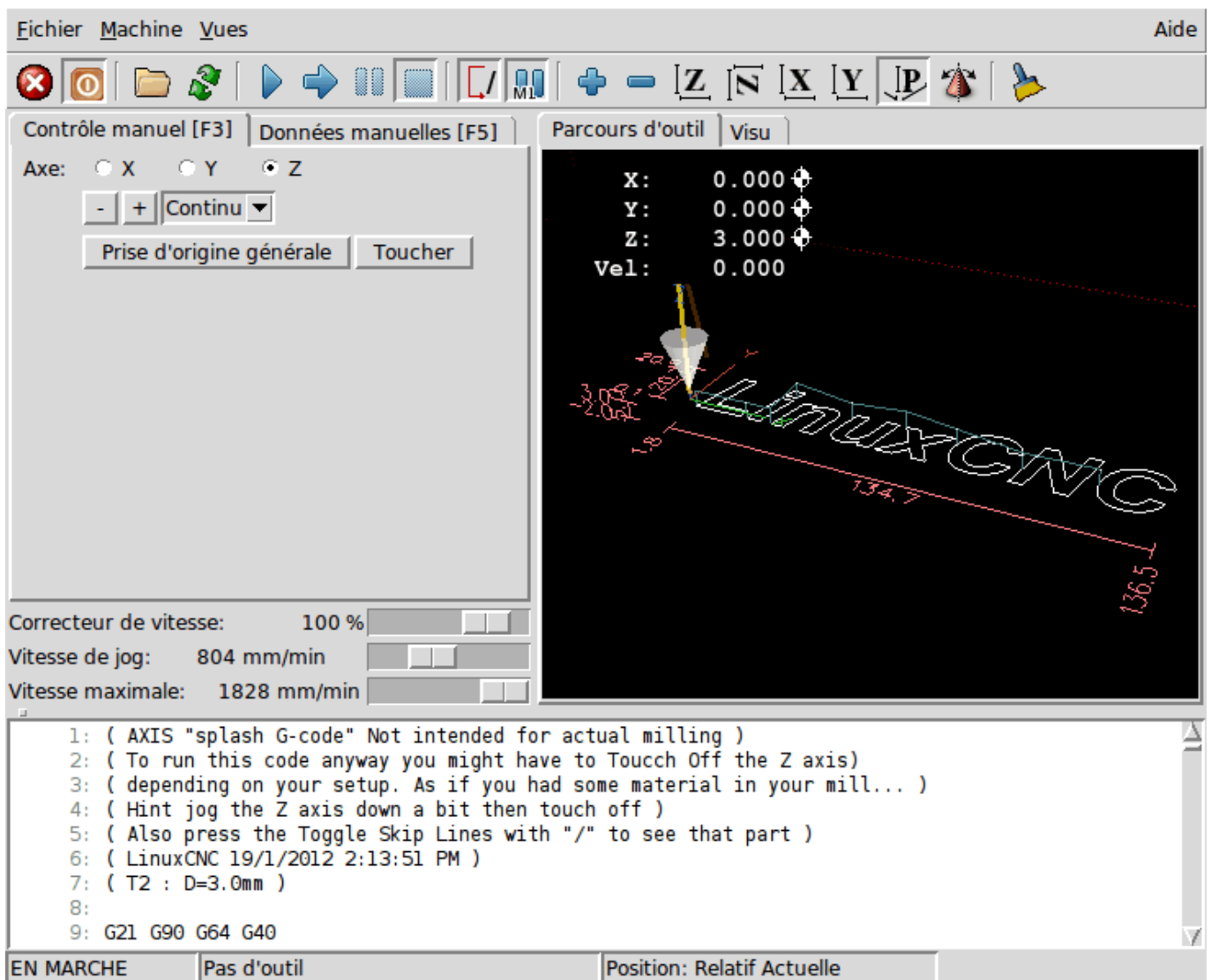


FIGURE 2.2 – L'interface graphique AXIS

- [Touchy](#), une interface graphique pour écran tactile.

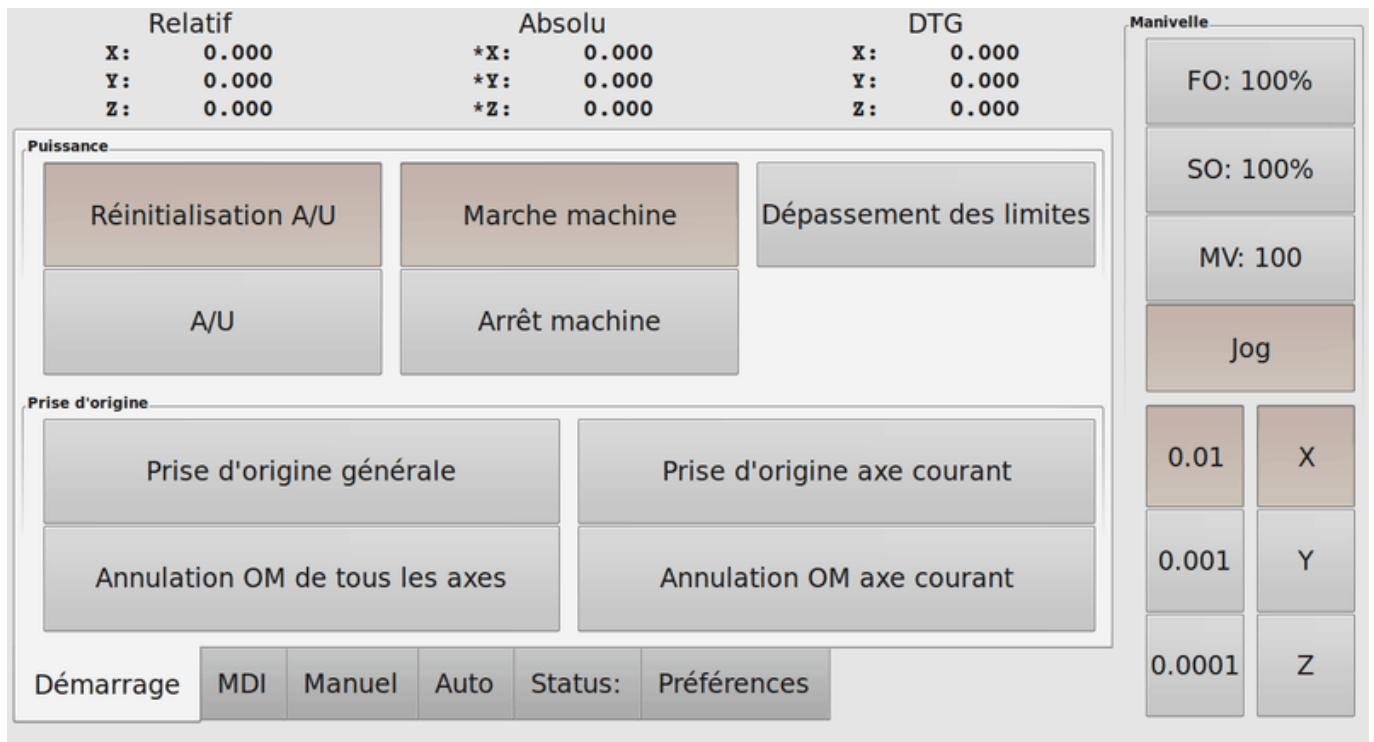


FIGURE 2.3 – L'interface graphique Touchy

- [NGCGUI](#), une interface graphique gérant les sous-programmes. Elle permet très simplement de créer des programmes G-code. Elle supporte surtout la concaténation de fichiers de sous-programmes, ce qui permet de construire des programmes G-code complets sans aucune programmation.

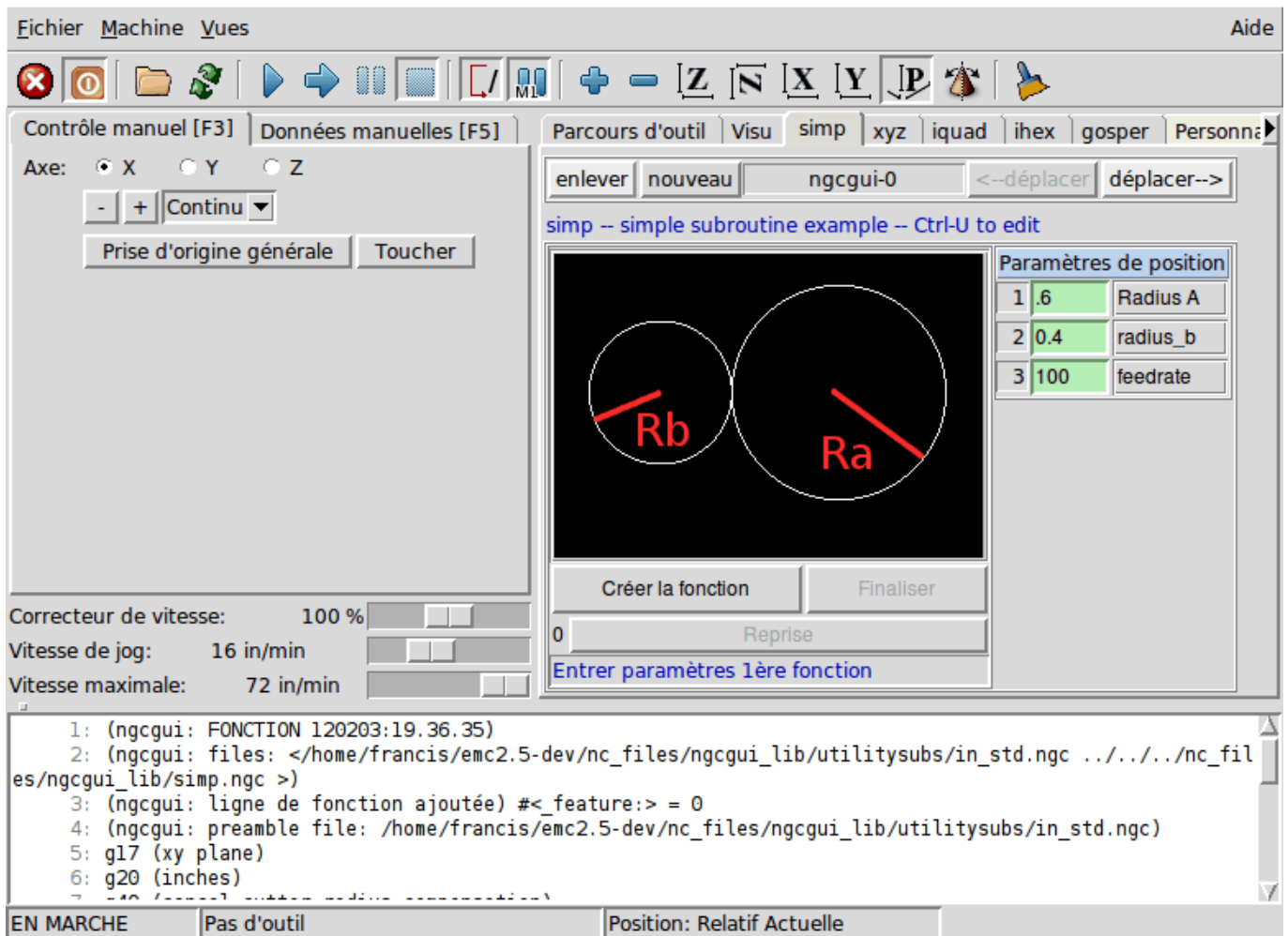


FIGURE 2.4 – L'interface graphique NGCGUI intégrée dans Axis

– [Mini](#), une interface en Tcl/Tk.

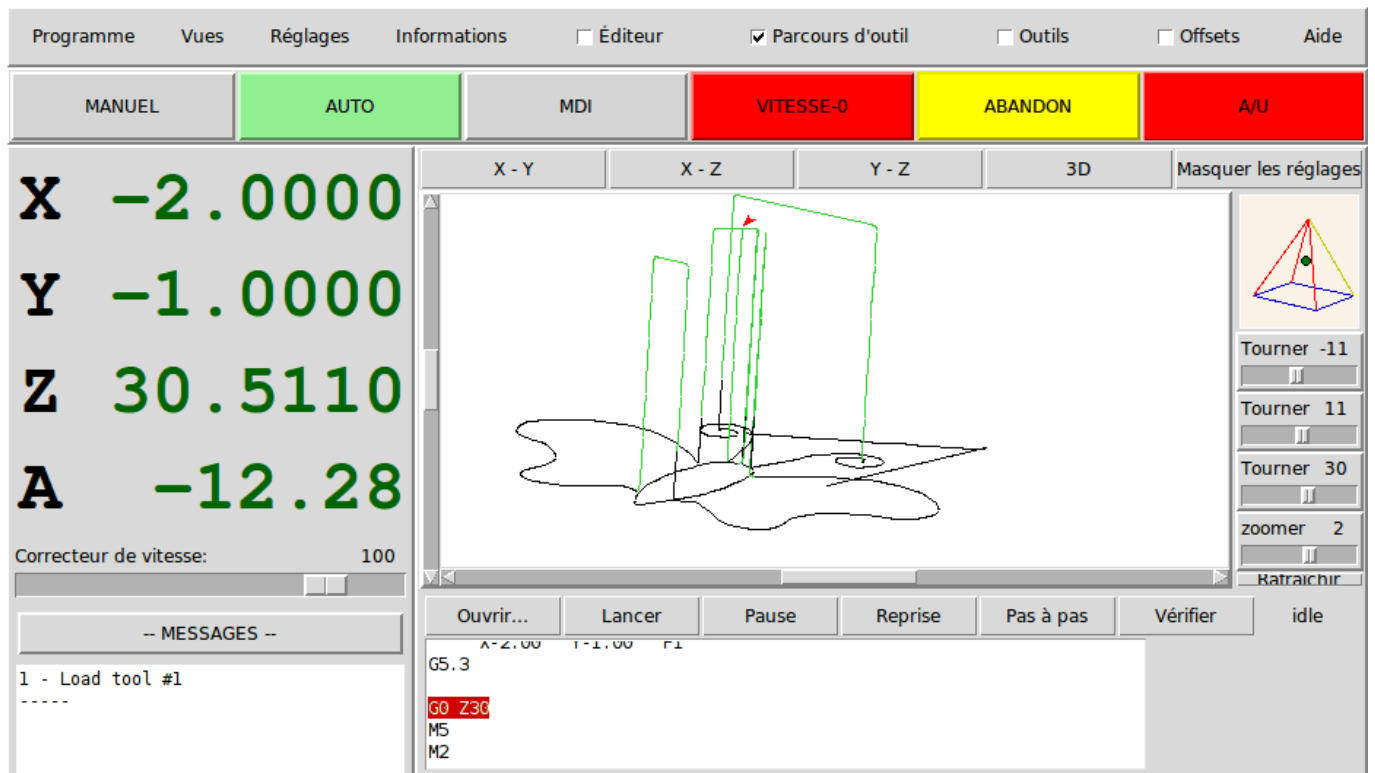


FIGURE 2.5 – L'interface graphique Mini

- [TkLinuxCNC](#), une autre interface basée sur Tcl/Tk. C'est l'interface la plus populaire après Axis

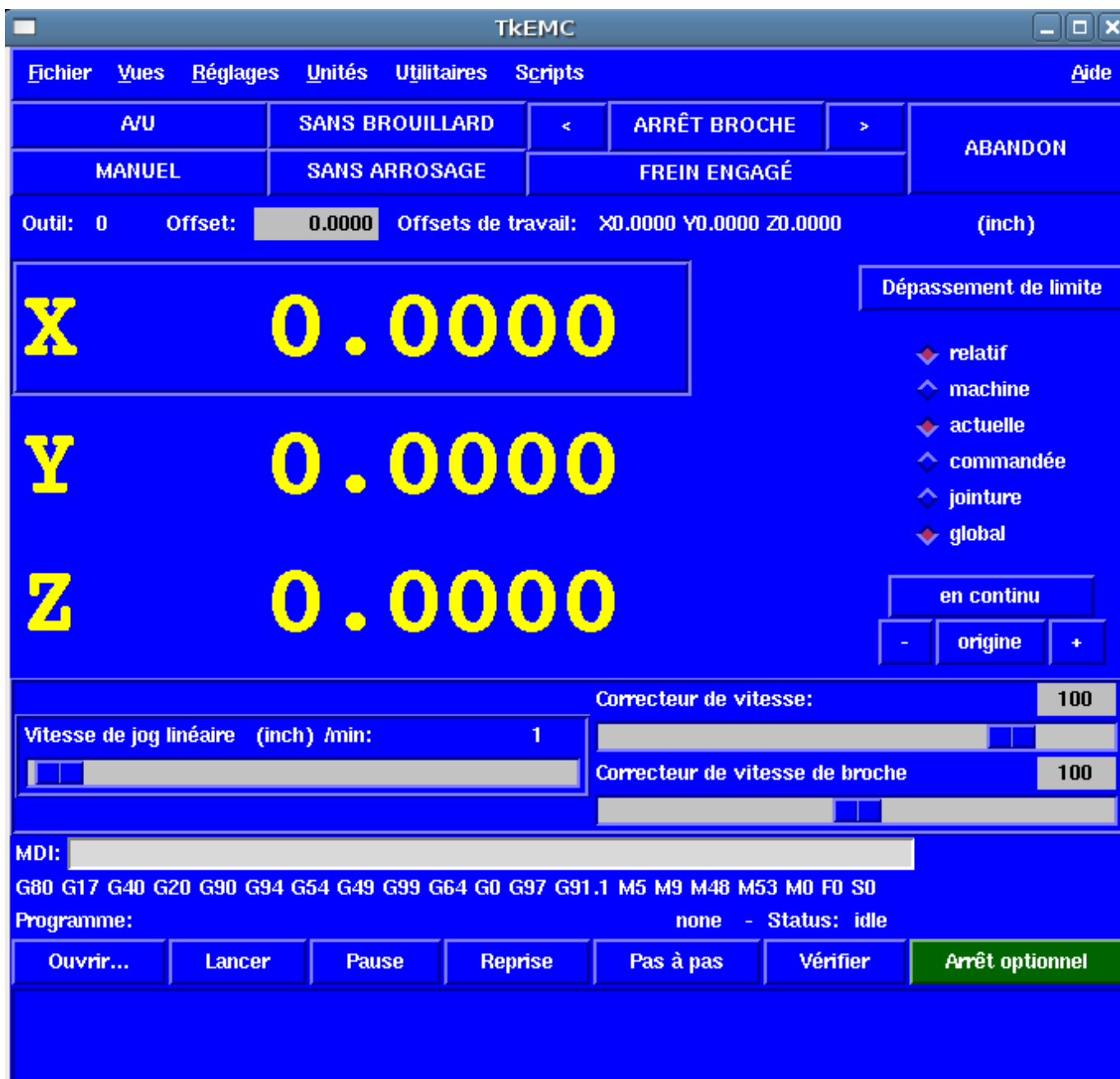


FIGURE 2.6 – L'interface graphique tklinuxcnc

- [Keystick](#), une interface graphique textuelle et minimaliste, appropriée pour les installations légères (sans serveur X par exemple).

```
F1 Estop On/Off      F5 MDI Mode          F9 Spndl Fwd/Off     ESC Aborts Actions
F2 Machine On/Off    F6 Reset Interp      F10 Spndl Rev/Off    TAB Selects Params
F3 Manual Mode       F7 Mist On/Off       F11 Spndl Decrease   END Quits Display
F4 Auto Mode         F8 Flood On/Off      F12 Spndl Increase   ? Toggles Help

  ESTOP
Override: 100%
Tool: 0
Offset: 0.0000

  MANUAL
LUBE OFF
LUBE OK

SPINDLE STOPPED
BRAKE ON
MIST OFF
FLOOD OFF

--- HOMED
X SELECTED
Speed: 60.0
Incr: continuous

Relative Act Pos:  --X--      --Y--      --Z--
                   0.0000    0.0000    0.0000
                   0.0000    0.0000    0.0000
```



FIGURE 2.7 – L’interface textuelle Keystick

- *Xemc*, un programme X-Windows
- *halui*, une interface utilisateur basée sur HAL, qui permet de contrôler LinuxCNC en utilisant des boutons et des interrupteurs
- *linuxcncrsh*, une interface utilisateur basée sur telnet, qui permet d’envoyer des commandes à partir d’ordinateurs distants de celui de LinuxCNC

## 2.4 Panneaux de contrôle virtuels

- *PyVCP*, un panneau de contrôle virtuel basé sur Python, il peut être intégré dans l’interface graphique Axis ou utilisé en autonome.
- *GladeVCP*, un panneau de contrôle virtuel basé sur Glade, il peut être intégré dans l’interface graphique Axis ou utilisé en autonome.

## 2.5 Langues

LinuxCNC utilise des fichiers traduits pour les interfaces utilisateur. Il fonctionne dans plusieurs langues et démarre dans la langue de la session ouverte par l’utilisateur au démarrage du PC. Si votre langue n’a pas encore été traduite contactez un développeur sur l’IRC ou sur la mailing liste si vous pouvez aider à la traduction.

## 2.6 Penser comme un opérateur sur CNC

Ce manuel ne prétend pas vous apprendre à utiliser un tour ou une fraiseuse. Devenir un opérateur expérimenté prends beaucoup de temps et demande beaucoup de travail. Un auteur a dit un jour, *Nous apprenons par l’expérience, si on la possède toute*. Les

outils cassés, les étaux attaqués et les cicatrices sont les preuves des leçons apprises. Une belle finition, des tolérances serrées et la prudence pendant le travail sont les preuves des leçons retenues. Aucune machine, aucun programme ne peut remplacer l'expérience humaine.

Maintenant que vous commencez à travailler avec le programme LinuxCNC, vous devez vous placer dans la peau d'un opérateur. Vous devez être dans le rôle de quelqu'un qui a la charge d'une machine. C'est une machine qui attendra vos commandes puis qui exécutera les ordres que vous lui donnerez. Dans ces pages, nous donnerons les explications qui vous aideront à devenir un bon opérateur de CNC avec LinuxCNC. Vous aurez besoin de bonnes informations ici, devant vous, c'est là que les pages suivantes prendront tout leur sens.

## 2.7 Modes opératoires

Quand LinuxCNC fonctionne, il existe trois différents modes majeurs pour entrer des commandes. Les modes *Manuel*, *Auto* et *MDI*. Passer d'un mode à un autre marque une grande différence dans le comportement de LinuxCNC. Des choses spécifiques à un mode ne peuvent pas être faites dans un autre. L'opérateur peut faire une prise d'origine sur un axe en mode manuel mais pas en mode auto ou MDI. L'opérateur peut lancer l'exécution complète d'un programme de G-codes en mode auto mais pas en mode manuel ni en MDI.

En mode manuel, chaque commande est entrée séparément. En termes humains, une commande manuelle pourrait être *active l'arrosage* ou *jog l'axe X à 250 millimètres par minute*. C'est en gros, équivalent à basculer un interrupteur ou à tourner la manivelle d'un axe. Ces commandes sont normalement contrôlées en pressant un bouton de l'interface graphique avec la souris ou en maintenant appuyée une touche du clavier. En mode auto, un bouton similaire ou l'appui d'une touche peut être utilisé pour charger ou lancer l'exécution complète d'un programme de G-codes stocké dans un fichier. En mode d'entrée de données manuelles (MDI) l'opérateur peut saisir un bloc de codes et dire à la machine de l'exécuter en pressant la touche *Return* ou *Entrée* du clavier.

Certaines commandes de mouvement sont disponibles et produisent les mêmes effets dans tous les modes. Il s'agit des commandes *Abandon*, *Arrêt d'Urgence* et *Correcteur de vitesse travail*. Ces commandes se dispensent d'explications.

L'interface utilisateur graphique AXIS supprime certaines distinctions entre Auto et les autres modes en rendant automatique la disponibilité des commandes, la plupart du temps. Il rend également floue la distinction entre Manuel et MDI parce que certaines commandes manuelles comme *Toucher*, sont également implémentées en envoyant une commande MDI. Il fait cela en changeant automatiquement le mode qui est nécessaire pour l'action que l'utilisateur a demandé.

# **Première partie**

## **Les interfaces utilisateur**

## Chapitre 3

# L'interface graphique AXIS

### 3.1 Introduction

AXIS est une interface utilisateur graphique pour LinuxCNC, il offre un aperçu permanent du parcours de l'outil. Il est écrit en Python, utilise Tk et OpenGL pour l'affichage de l'interface graphique.

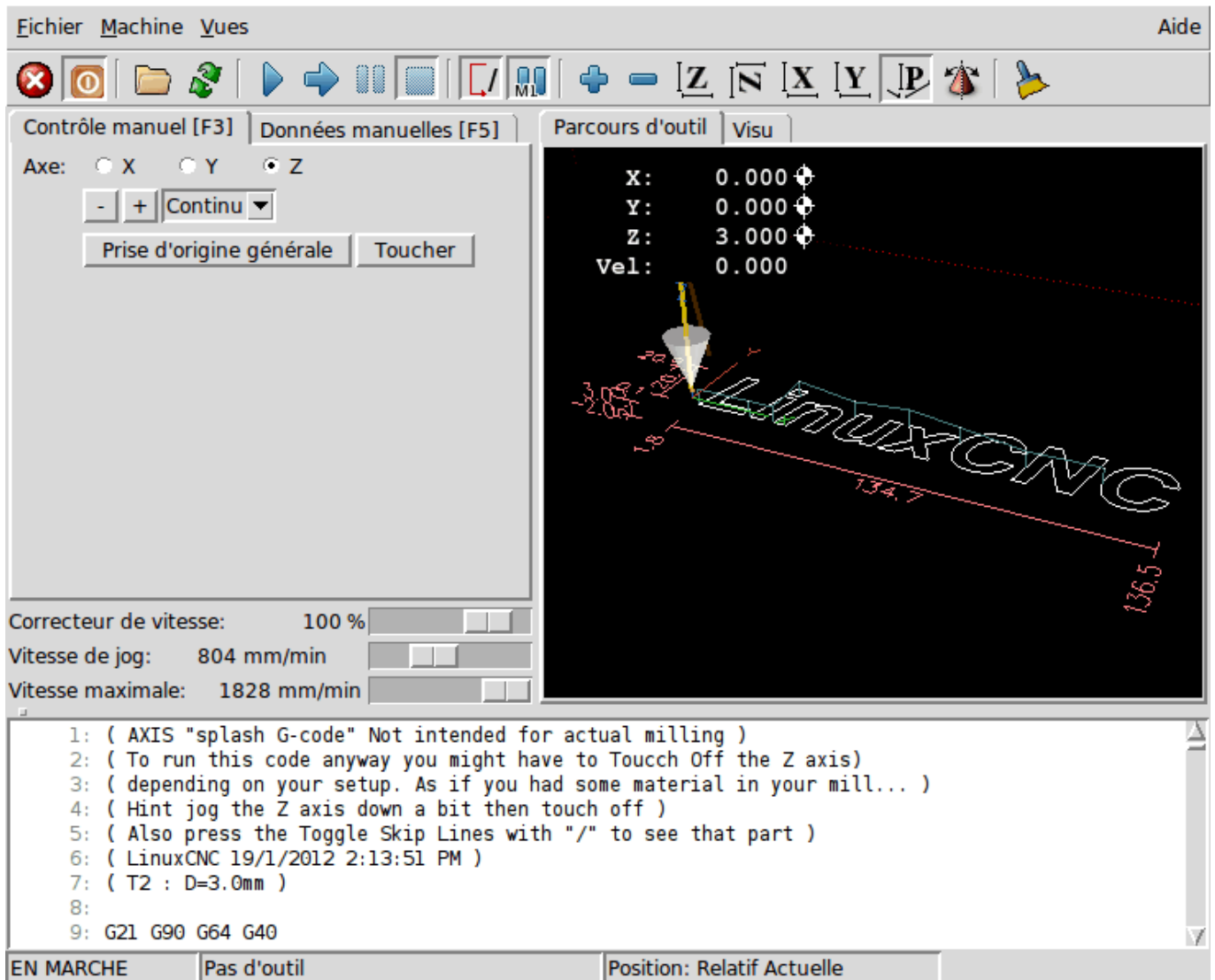


FIGURE 3.1 – Fenêtre d'AXIS

## 3.2 Commencer avec AXIS

Pour choisir AXIS comme interface graphique de LinuxCNC, éditez le fichier .ini et dans la section [DISPLAY] changez la ligne DISPLAY comme ceci:

```
DISPLAY = axis
```

Puis, lancez LinuxCNC et choisissez le fichier ini. La configuration simplifiée *sim/axis.ini* est déjà configurée pour utiliser AXIS comme interface graphique.

Quand AXIS démarre, une fenêtre telle que celle de la figure [ci-dessus](#) s'ouvre.

### 3.2.1 Une session typique avec AXIS

1. Lancer LinuxCNC et sélectionner un fichier de configuration.
2. Relâcher le bouton d'arrêt d'urgence *A/U* et presser celui de *Marche* machine.

3. Faire les prises d'origine machine *POM* pour chacun des axes.
4. Charger un fichier d'usinage.
5. Utiliser l'affichage du parcours d'outil pour vérifier que le programme est correct.
6. Brider le brut à usiner sur la table.
7. Faire les prises d'origine pièce *POP* de chacun des axes avec le jog et en utilisant le bouton *Toucher*.
8. Lancer le programme.
9. Pour usiner le même fichier une nouvelle fois, retourner à l'étape 6. Pour usiner un fichier différent, retourner à l'étape 4. Quand le travail est terminé, quitter AXIS.

### 3.3 Éléments de la fenêtre d'AXIS

La fenêtre d'AXIS contient les éléments suivants:

- Un espace d'affichage qui montre une pré-visualisation du fichier chargé (dans ce cas, *axis.ngc*), ainsi que la position courante du point programmé par la machine. Plus tard, cette zone affichera le parcours de l'outil déplacé par la machine, cette zone est appelée le parcours d'outil (backplot)
- Une barre de menus, une barre d'outils, des curseurs et des onglets permettant d'effectuer différentes actions.
- L'onglet *Contrôle manuel*, qui permet de faire des mouvements d'axe, de mettre la broche en rotation ou de l'arrêter, de mettre l'arrosage en marche ou de l'arrêter.
- L'onglet *Données manuelles* (appelé aussi MDI), où les blocs de programme G-code peuvent être entrés et exécutés à la main, une ligne à la fois.
- Les curseurs *Correcteurs de vitesse*, qui permettent d'augmenter ou de diminuer la vitesse de la fonction concernée.
- Une zone de textes qui affiche le G-code du fichier chargé.
- Une barre d'état qui affiche l'état de la machine. Dans cette capture d'écran, la machine est en marche, aucun outil n'est monté, la position affichée est *relative* à l'origine machine (par opposition à une position *absolue*) et *actuelle* (par opposition à une position *commandée*)

#### 3.3.1 Options des menus

Certaines options de menu peuvent s'afficher en grisé, c'est dépendant des options du fichier de configuration ini.

##### 3.3.1.1 Menu Fichier

###### Ouvrir...

C'est une boîte de dialogue standard pour ouvrir un fichier G-code à charger dans AXIS. Si un filtre de programme a été configuré, il peut aussi être ouvert ici.

###### Fichiers récents...

Affiche la liste des fichiers ouverts récemment.

###### Éditer...

Ouvre le fichier G-code courant pour édition si un éditeur a été déclaré dans le fichier ini.

###### Recharger...

Recharge le fichier G-code courant. Si le fichier a été édité, il doit être rechargé pour que les modifications prennent effet. Si un programme a été stoppé, pour le reprendre depuis le début, le recharger. Le bouton *Recharger* a le même effet que l'option de menu.

###### Enregistrer le G-code sous...

Enregistre le fichier courant sous un nouveau nom.

###### Propriétés...

Donne la somme des mouvements en vitesse rapide et celle en vitesse travail. Ne tient pas compte des accélérations, ni des décélérations, ni des modes de trajectoire, de sorte qu'il ne donne jamais de temps inférieur au temps réel d'exécution.

###### Éditer la table d'outils...

Ouvre un dialogue permettant d'éditer les valeurs de la table d'outils.

### **Recharger la table d'outils...**

Après avoir édité la table d'outil, il convient de la recharger pour que les nouvelles valeurs soient prises en compte.

### **Éditeur de Ladder...**

Si Classic Ladder a été chargé, il est possible de l'éditer ici.

### **Quitter...**

Termine la session courante de LinuxCNC.

## **3.3.1.2 Menu Machine**

### **Arrêt d'Urgence F1...**

(bascule) Active/désactive l'arrêt d'urgence.

### **Marche/Arrêt F2...**

(bascule) Active/désactive la puissance machine.

### **Démarrer le programme...**

Lance l'exécution du programme G-code.

### **Démarrer à la ligne sélectionnée...**

Prudence avec cette commande, respecter la démarche suivante: Premièrement, sélectionner à la souris, la ligne à laquelle démarrer. Déplacer ensuite manuellement, l'outil à la position de la ligne précédente puis, cette commande exécutera le reste du code.

### **Pas à pas...**

Avance d'un seul pas de programme.

### **Pause...**

Effectue une pause dans le programme.

### **Reprise...**

Reprends la marche après une pause.

### **Stopper...**

Stoppe le programme en marche.

### **Arrêt sur M1...**

Si un M1 est rencontré et que cette option est cochée, l'exécution du programme s'interrompra à la ligne du M1. Presser *Reprise* pour continuer.

### **Sauter les lignes avec "/"...**

Si une ligne commençant par / est rencontrée et que cette option est cochée, cette ligne sera sautée.

### **Vider l'historique du MDI...**

Efface l'historique des données manuelles.

### **Copier depuis l'historique du MDI...**

Copier l'historique des données manuelles dans le presse-papier.

### **Coller dans l'historique du MDI...**

Coller le contenu du presse-papier dans la fenêtre d'historique des données manuelles.

### **Calibration...**

Lance un assistant de réglage de PID. Cette option n'est utile qu'avec un système à servomoteurs.

### **Afficher configuration de HAL...**

Ouvre une fenêtre sur la configuration de HAL depuis laquelle il est possible de visualiser tous les *Components*, *Pins*, *Parameters*, *Signals*, *Functions* et *Threads* de HAL.

### **HAL Mètre...**

Ouvre une fenêtre dans laquelle il est possible de visualiser un seul *Signal*, *HAL Pin*, ou *Parameter* de HAL.

### **HAL Scope...**

Ouvre un oscilloscope virtuel qui permet de tracer dans le temps, les valeurs de HAL.

### **Afficher l'état de LinuxCNC...**

Ouvre une fenêtre montrant l'état de LinuxCNC.

### **Choisir le niveau de Debug...**

Ouvre une fenêtre dans laquelle les niveaux de débogage sont visibles et certains réglables.

#### Prise d'origine...

Effectue la prise d'origine machine d'un ou de tous les axes.

#### Annulation OM...

Annule les origines d'un ou de tous les axes.

#### Annulation décalage d'origine...

Annule les décalages d'origine du système de coordonnées choisi.

#### L'outil touchera la pièce...

Lorsqu'un *Toucher* est effectué, la valeur entrée est relative au système de coordonnées pièce actuel (G5x), tel que modifié par le décalage d'axe (G92). Quand la séquence de *Toucher* est complète, la coordonnée relative pour l'axe choisi prendra la valeur entrée. (Voir aussi G10 L10)

#### L'outil touchera le porte-pièce...

Lorsqu'un *Toucher* est effectué, la valeur entrée est relative au 9ème système de coordonnées (G59.3), le décalage d'axe (G92) est ignoré. Mode destiné aux machines possédant un porte-pièce référencé à un endroit, sur lequel s'effectue le *Toucher*. Le 9ème système de coordonnées doit être ajusté pour que la pointe d'un outil de longueur nulle (le nez de broche), soit à l'origine du porte-pièce quand les coordonnées relatives sont à 0. Voir aussi [G10 L11](#).

### 3.3.1.3 Menu Vues

#### Tout est dans le point de vue...

Les icônes de choix du type d'affichage et du menu *Vues* d'AXIS se réfèrent à des *Vue de dessus*, *Vue de face* et *Vue de côté*. Ces termes sont corrects si la machine CNC a un axe Z vertical, avec une valeur de Z positive en haut. C'est vrai pour les fraiseuses verticales, qui sont probablement les plus populaires, c'est également vrai pour toutes les machines d'électro-érosion et aussi les tours verticaux, sur lesquels la pièce tourne sous l'outil.

Les termes *Vue de dessus*, *Vue de face* et *Vue de côté* sont cependant source de confusion sur d'autres machines CNC, comme un tour standard, sur lequel l'axe Z est horizontal, ou sur une fraiseuse horizontale, qui a également l'axe Z horizontal, ou même un tour vertical inversé, sur lequel la pièce tourne au dessus de l'outil et qui a son axe Z positif vers le bas!

Il faut juste se rappeler que l'axe Z est toujours parallèle à la broche et plus positif en s'éloignant de celle-ci. Soyez familiarisé avec la cinématique de vos machines, vous interpréterez alors l'affichage comme il se doit.

#### Vue de dessus...

La vue de dessus (ou vue de Z) affiche l'aspect du G-code vu depuis le côté positif de l'axe Z et en regardant vers son côté négatif. Cette vue convient bien pour visualiser les axes X et Y.

#### Vue de dessus basculée...

La vue de dessus basculée (ou vue de Z basculé) affiche également l'aspect du G-code vu depuis le côté positif de l'axe Z et en regardant vers son côté négatif. Mais cette fois, les axes X et Y sont représentés pivotés de 90 degrés pour mieux occuper l'espace d'affichage. Cette vue convient bien également, pour visualiser les axes X et Y.

#### Vue de côté...

La vue de côté (ou vue de X) affiche l'aspect du G-code vu depuis le côté positif de l'axe X et en regardant vers son côté négatif. Cette vue convient pour visualiser les axes Y et Z.

#### Vue de face...

La vue de face (ou vue de Y) affiche l'aspect du G-code vu depuis le côté positif de l'axe Y et en regardant vers son côté négatif. Cette vue convient bien pour visualiser les axes X et Z.

#### Vue en perspective...

La vue en perspective (ou vue P) affiche l'aspect du G-code en regardant vers la pièce depuis un point de vue orientable, par défaut vers X+, Y-, Z+. Cette position est orientable en la sélectionnant à la souris. L'affichage est un compromis, il tente d'afficher en 3D, entre trois et neuf axes, sur un écran en deux dimensions. Il y aura donc souvent certaines caractéristiques difficiles à voir, ce qui requerra un changement de point de vue. Cette vue convient bien pour voir les trois axes à la fois.

#### Affichage en pouces...

Ajuste l'échelle d'affichage d'AXIS pour les pouces.

#### Affichage en mm...

Ajuste l'échelle d'affichage d'AXIS pour les millimètres.

### **Afficher le programme...**

L'affichage à l'écran de l'aspect du G-code peut être entièrement désactivé si l'opérateur le souhaite.

### **Parcours d'outil en vitesse rapide...**

L'affichage du parcours d'outil du programme G-code courant représente toujours les mouvements en vitesse travail (G1,G2,G3) en blanc. Mais l'affichage des mouvements en vitesse rapide (G0) en cyan peut être désactivé si l'opérateur le souhaite.

### **Simulation de transparence...**

Cette option rends plus lisible le tracé des parcours affichés par les programmes complexes, mais il peut rendre l'affichage plus lent.

### **Parcours d'outil en temps réel...**

La surbrillance des chemins d'outils en vitesse travail (G1,G2,G3) quand l'outil se déplace peut être désactivée si l'opérateur le souhaite.

### **Afficher l'outil...**

Le symbole d'un outil, représenté par un cône ou un cylindre peut être désactivé si l'opérateur le souhaite.

### **Afficher les étendues...**

L'affichage des étendues du programme G-code chargé (déplacements maximum de chacun des axes), peut être désactivé si l'opérateur le souhaite.

### **Afficher les offsets...**

L'emplacement de l'origine du système de coordonnées pièce (G54 à G59.3) peut être représenté par un jeu de trois lignes orthogonales, une rouge, une bleue et une verte. L'affichage de cette origine pièce (ou zéro pièce), peut être désactivé si l'opérateur le souhaite.

### **Afficher les limites machine...**

Les limites maximales de déplacement machine pour chacun des axes, qui sont fixées dans le fichier ini, s'affichent comme une boîte rectangulaire en lignes pointillées rouges. Il est facile, au chargement d'un nouveau programme G-code, de voir si la pièce est contenue dans le volume représenté. Ou de vérifier de combien l'étau doit être décalé, pour que le G-code puisse être usiné sans dépasser les limites de déplacements de la machine. Cette option peut être désactivée si l'opérateur le souhaite.

### **Afficher la vitesse d'avance...**

L'affichage de la vitesse peut être utile pour voir la précision avec laquelle la machine suit la vitesse commandée. Cette option peut être désactivée si l'opérateur le souhaite.

### **Afficher la distance restante...**

La distance restante est une valeur très utile à suivre, au lancement d'un programme de G-code inconnu pour la première fois. En combinaison avec les curseurs des correcteurs de vitesse, des dégâts sur l'outil ou la machine peuvent être évités. Quand le programme G-code sera débogué et qu'il fonctionnera en douceur, l'affichage de la distance restante pourra être désactivée si l'opérateur le souhaite.

### **Coordonnées en police large...**

Les coordonnées des axes et la vitesse d'avance, s'afficheront en police large dans la vue du parcours d'outil.

### **Rafraîchir le parcours d'outil...**

Au fur et à mesure des déplacements de l'outil, les parcours s'affichent sur l'écran d'Axis en surbrillance. Avant de répéter le programme, ou pour avoir un affichage clair sur une zone intéressante, la surbrillance des parcours précédents peut être rafraîchie.

### **Afficher la position commandée...**

C'est la position que LinuxCNC cherche à atteindre. Quand le mouvement est stoppé, c'est la position que LinuxCNC cherchera à maintenir.

### **Afficher la position actuelle...**

La position actuelle est la position mesurée grâce aux informations issues des codeurs ou simulées par le générateur de pas. Elle peut différer légèrement de la position commandée pour diverses raisons, comme les réglages des boucles PID, les contraintes physiques ou les efforts de coupe.

### **Afficher la position machine...**

C'est la position par rapport à l'origine machine, telle qu'établie par la prise d'origine machine (*POM*).

### **Afficher la position relative...**

C'est la position par rapport à l'origine pièce, telle qu'établie par la prise d'origine pièce (*POP*). On peut aussi représenter cette position comme étant l'origine machine à laquelle on a appliqué les codes de décalages des systèmes de coordonnées G5x, G92 et G43.

### 3.3.1.4 Menu Aide

#### A propos d'Axis...




















Donne la version et quelques informations relatives au copyright.

#### Aide rapide...

Affiche la liste des raccourcis clavier.

### 3.3.2 Boutons de la barre d'outils

Signification des boutons de la fenêtre d'AXIS, de gauche à droite:

1.  Arrêt d'urgence (A/U) (en Anglais, E-Stop)
2.  Marche/Arrêt puissance machine
3.  Ouvrir un fichier
4.  Recharger le fichier courant
5.  Départ cycle
6.  Cycle en pas à pas
7.  Pause/Reprise
8.  Stopper l'exécution du programme
9.  Sauter ou non les lignes commençant par /
10.  Avec ou sans pause optionnelle
11.  Zoom plus
12.  Zoom moins
13.  Vue prédéfinie **Z** (vue de dessus)
14.  Vue prédéfinie **Z basculée**
15.  Vue prédéfinie **X** (vue de côté)
16.  Vue prédéfinie **Y** (vue de face)
17.  Vue prédéfinie **P** (vue en perspective)
18.  Orienter la vue avec le bouton gauche de la souris
19.  Rafraîchir le parcours d'outil

### 3.3.3 Zones d'affichage graphique du programme

#### 3.3.3.1 Affichage des coordonnées

L'affichage des coordonnées est situé en haut à gauche de l'écran graphique. Il montre les positions de la machine. A gauche du nom de l'axe, un symbole d'origine est visible si la prise d'origine de l'axe a été faite.



Symbole de prise d'origine faite.

A droite du nom de l'axe, un symbole de limite est visible si l'axe est sur un de ses capteurs de limite.



Symbole de limite d'axe.

Pour interpréter correctement ces valeurs, référez vous à l'indicateur *Position* de la barre d'état. Si la position est *Absolue*, alors les valeurs affichées sont exprimées en coordonnées machine. Si la position est *Relative*, alors les valeurs affichées sont exprimées en coordonnées relatives à la pièce. Quand les coordonnées affichées sont relatives, une marque d'origine de couleur cyan est visible pour représenter l'origine machine.



Symbole d'origine machine.

Si la position est *Commandée*, alors il s'agit de la position à atteindre. Par exemple, les coordonnées passées dans une commande **G0**. Si la position est *Actuelle*, alors il s'agit de la position à laquelle la machine vient de se déplacer. Ces valeurs peuvent varier pour certaines raisons: erreur de suivi, bande morte, résolution d'encodeur, ou taille de pas. Par exemple, si vous demandez un mouvement à X 0.08 à votre fraiseuse, mais un pas du moteur fait 0.03, alors la position *Commandée* sera de 0.08, mais la position *Actuelle* sera de 0.06 (2 pas) ou 0.09 (3 pas).

#### 3.3.3.2 Vue du parcours d'outil

Quand un fichier est chargé, une vue du parcours d'outil qu'il produira est visible dans la zone graphique. Les mouvements en vitesse rapide (tels ceux produits par une commande **G0**) sont affichés en lignes pointillées vertes. Les déplacements en vitesse travail (tels ceux produits par une commande **G1**) sont affichés en lignes continues blanches. Les arrêts temporisés (tels ceux produits par la commande **G4**) sont représentés par une petite marque **X**.

Un mouvement G0 (Vitesse rapide) avant un déplacement en vitesse travail ne sera pas affiché sur l'écran des parcours d'outil. Un mouvement en vitesse rapide, après un appel d'outil T<n>, n'apparaîtra sur l'écran des parcours d'outil qu'après le mouvement en vitesse travail suivant. Pour contourner une de ces caractéristiques, programmer un G1 sans déplacement, juste avant le G0.

#### 3.3.3.3 Étendues du programme

Les *étendues* du programme sont affichées pour chacun des axes. Aux extrémités, les coordonnées minimales et maximales sont indiquées. Au centres, la différence, entre ces deux coordonnées, est indiquée.

Quand une coordonnée dépasse la limite logicielle fixée dans le fichier .ini, la coordonnée correspondante s'affiche en rouge, entourée d'un rectangle. Dans la figure ci-dessous, la limite maximale est dépassée sur l'axe X, comme l'indique le rectangle entourant la valeur de la coordonnée. Le déplacement X minimal du programme est de -1.95, la course maximale est de 1.88 en X et le programme nécessite un déplacement en X de 3.83 pouces. Le déplacement total demandé par le programme est donc possible. Pour cela, se déplacer en jog vers la gauche puis *Toucher* à nouveau pour corriger l'origine pièce.



FIGURE 3.2 – Limites logicielles

#### 3.3.3.4 Le cône d'outil

Si aucun outil n'est chargé, l'emplacement de la pointe de l'outil est indiqué par le *cône d'outil*. Le cône d'outil ne donne aucune indication sur la forme, la longueur, ou le rayon de l'outil.

Quand un outil est chargé, par exemple dans le MDI, avec la commande **T1 M6**, le cône d'outil passe de conique à cylindrique, il indique alors la proportion du diamètre de l'outil lu dans le fichier de la table d'outils.

#### 3.3.3.5 Parcours d'outil

Quand la machine se déplace, elle laisse une trace appelée le parcours d'outil. La couleur des lignes indique le type de mouvement: jaune pour les mouvements jog, vert clair pour les mouvements en vitesse rapide, rouge pour les mouvements en vitesse d'avance programmée et magenta pour les mouvements circulaires en vitesse d'avance programmée.

#### 3.3.3.6 Interaction avec l'affichage

Par un clic gauche sur une portion du parcours d'outil, la ligne sous la souris passe en surbrillance à la fois dans le parcours d'outil et dans le texte. Un clic droit dans une zone vide enlève la surbrillance

En déplaçant la souris avec son bouton gauche appuyé, la vue est glissée sur l'écran.

En déplaçant la souris avec le bouton *Maj* enfoncé, ou en glissant avec la molette de la souris appuyée, la vue est tournée. Si une ligne du tracé est en surbrillance, elle devient le centre de rotation de la vue. Autrement, le centre de rotation est le milieu du fichier dans son ensemble.

En tournant la molette de la souris, ou en glissant la souris avec son bouton droit enfoncé, ou encore en glissant la souris avec son bouton gauche enfoncé et la touche *Ctrl* appuyée, le tracé sera zoomé en plus ou en moins.

En cliquant sur une des icônes de vue pré-définie de la barre d'outils, ou en pressant la touche **V**, cette vue est sélectionnée.

### 3.3.4 Zone texte d'affichage du programme

Un clic gauche sur une ligne du programme passe la ligne en surbrillance à la fois dans la zone texte et dans le parcours d'outil.

Quand le programme est lancé, la ligne en cours d'exécution est en surbrillance rouge. Si aucune ligne n'est sélectionnée par l'utilisateur, le texte défile automatiquement pour toujours laisser la ligne courante visible.

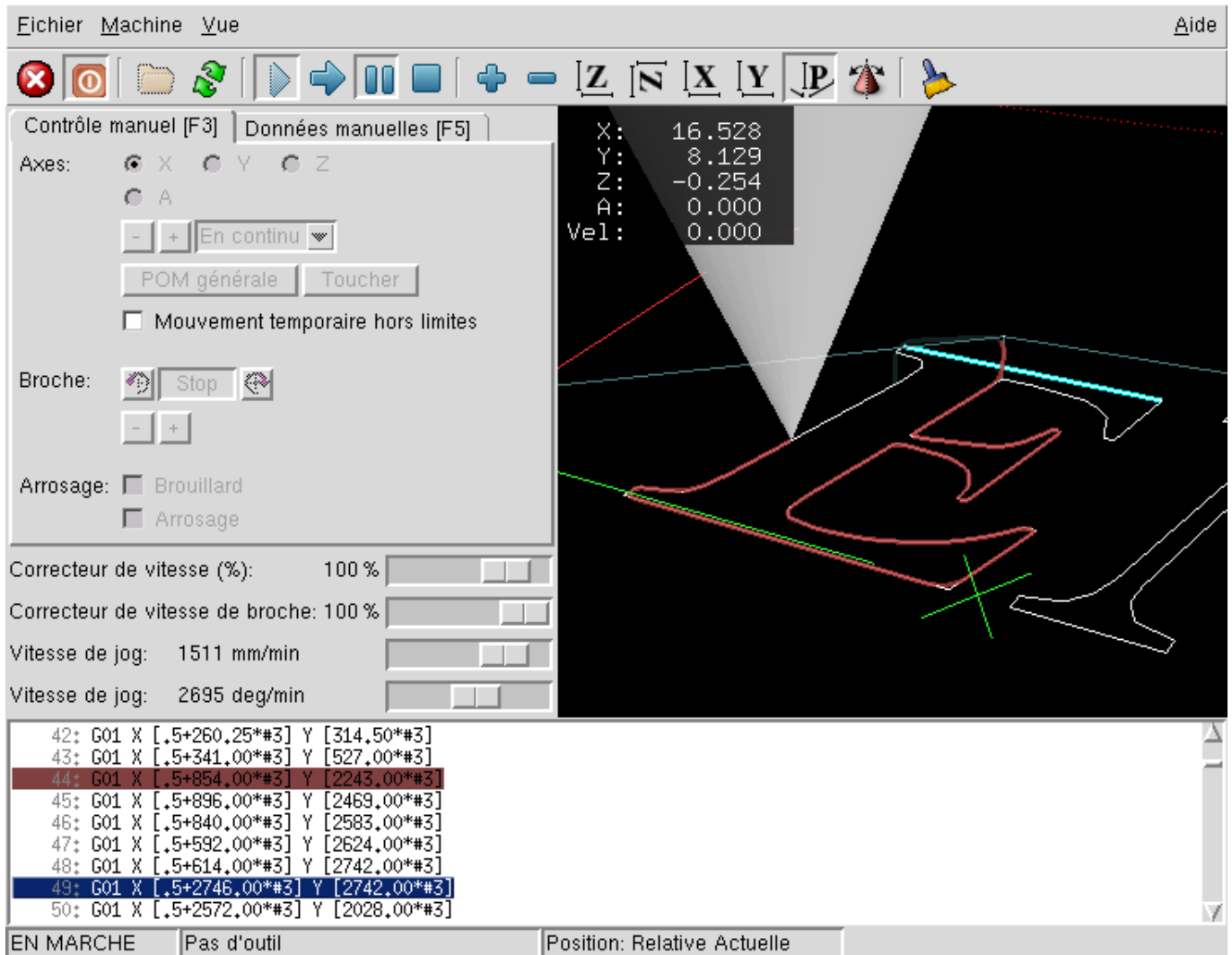


FIGURE 3.3 – Ligne courante et ligne en surbrillance

### 3.3.5 Contrôle manuel

Quand la machine est en marche mais qu'aucun programme n'est exécuté, les éléments graphiques de l'onglet *Contrôle manuel* peuvent être utilisés pour actionner la machine ou mettre en marche et arrêter ses différents organes.

Quand la machine n'est pas en marche, ou quand un programme est en cours d'exécution, le contrôle manuel n'est pas disponible.

Certains des éléments décrits plus bas ne sont pas disponibles sur toutes les machines. Quand AXIS détecte qu'une pin particulière n'est pas connectée dans le fichier HAL, l'élément correspondant de l'onglet *Contrôle manuel* est supprimé. Par exemple, si la pin HAL *motion.spindle-brake* n'est pas connectée, alors le bouton *Frein de broche* n'apparaîtra pas sur l'écran. Si la variable d'environnement *AXIS\_NO\_AUTOCONFIGURE* est mise à 1, ce comportement est désactivé et tous les boutons sont visibles.

#### 3.3.5.1 Le groupe de cases et boutons Axes

Les cases à cocher du groupe *Axes* permettent de choisir l'axe de la machine à actionner manuellement. Cette action s'appelle le *jog*. Premièrement sélectionner l'axe à actionner en cochant sa case. Puis cliquer sur le bouton + ou - selon le sens de déplacement souhaité. Les quatre premiers axes peuvent aussi être déplacés avec les touches fléchées pour X et Y, avec les touches Page précédente et Page suivante pour (Z) et les touches [ et ] pour A.

Si *En continu* est sélectionné, le mouvement continuera tant que la touche ou le bouton resteront appuyés. Si une autre valeur est sélectionnée, la machine se déplacera juste de la distance affichée à chaque fois que la touche ou le bouton seront appuyés. Par défaut, les valeurs disponibles sont:

0.1000 0.0100 0.0010 0.0001

Voir le Manuel de l'intégrateur pour plus d'informations sur la configuration des incréments de jog.

### 3.3.5.2 La prise d'origine machine

Si votre machine dispose de contacts d'origine machine et a une séquence de prise d'origine définie dans le fichier ini, le bouton *POM générale* lancera cette séquence pour tous les axes, les touches *Ctrl-HOME* auront le même effet.

Si votre machine dispose de contacts d'origine mais n'a pas de séquence de prise d'origine définie dans le fichier ini, le bouton *POM générale* effectuera uniquement la prise d'origine de l'axe sélectionné. Cette procédure doit alors être réalisée, séparément pour chacun des axes.

Si votre machine ne dispose d'aucun contact d'origine défini dans la configuration, le bouton *POM générale* définira la position actuelle de l'axe comme étant la position d'origine machine et l'axe sera marqué comme ayant sa prise d'origine machine faite.

### 3.3.5.3 Toucher

Si le bouton *Toucher* ou la touche *FIN* sont appuyés, le décalage d'origine pièce de l'axe Z, sur la figure ci-dessous: P1 G54, prendra la valeur spécifiée dans le champ de la boîte de dialogue. Les expressions peuvent être entrées en suivant les règles de programmation rs274ngc, sauf les variables qui ne peuvent pas être utilisées. La valeur résultante sera affichée sous le champ. Exemple, pour faire la prise d'origine pièce, on affleure l'outil sur une cale de 5mm d'épaisseur posée sur le bloc, on presse le bouton *Toucher* et on saisi 5 dans le champ de la boîte de dialogue. La pointe de l'outil sera alors référencée à 0 sur la surface du bloc.

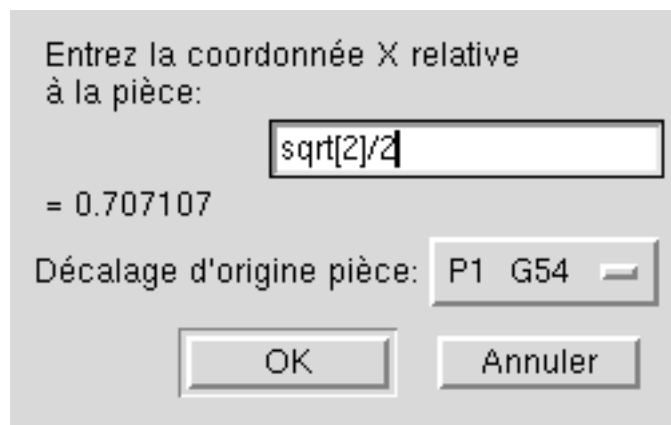


FIGURE 3.4 – Fenêtre du Toucher

Voir aussi les options du menu Machine: *Toucher la pièce* et *Toucher le porte-pièce*.

### 3.3.5.4 Dépassement de limite

En appuyant sur *Dépassement de limite*, la machine sera temporairement autorisée à se déplacer au delà d'un contact de limite physique. Cette case à cocher n'est disponible que lorsqu'un fin de course est pressé. Elle est désactivée après chaque mouvement de jogging. Si l'axe est configuré avec des contacts positifs et négatifs séparés, LinuxCNC permettra le jogging uniquement dans le sens du dégagement. *Dépassement de limite* ne permettra pas un jogging au delà d'une limite logicielle. La seule façon de désactiver une limite logicielle sur un axe est d'annuler sa prise d'origine.

### 3.3.5.5 Le groupe *Broche*

Les boutons de la première rangée permettent de sélectionner la direction de rotation de la broche: Sens anti-horaire, Arrêt, Sens horaire. Les boutons de la rangée suivante augmentent ou diminuent la fréquence de rotation. La case à cocher de la troisième rangée permet d'engager ou de relâcher le frein de broche. Selon la configuration de votre machine, ces éléments n'apparaîtront peut être pas tous.

### 3.3.5.6 Le groupe *Arrosage*

Ces deux boutons permettent d'activer les *gouttelettes* et l'*Arrosage fluide* ou de les désactiver. Selon la configuration de votre machine, ces boutons n'apparaîtront peut être pas tous.

## 3.3.6 Données manuelles (MDI)

L'onglet d'entrée de données manuelles (encore appelé MDI), permet d'entrer et d'exécuter manuellement et une par une, des lignes de programme en G-code. Quand la machine n'est pas en marche, ou quand un programme est en cours d'exécution, cet onglet n'est pas opérationnel.

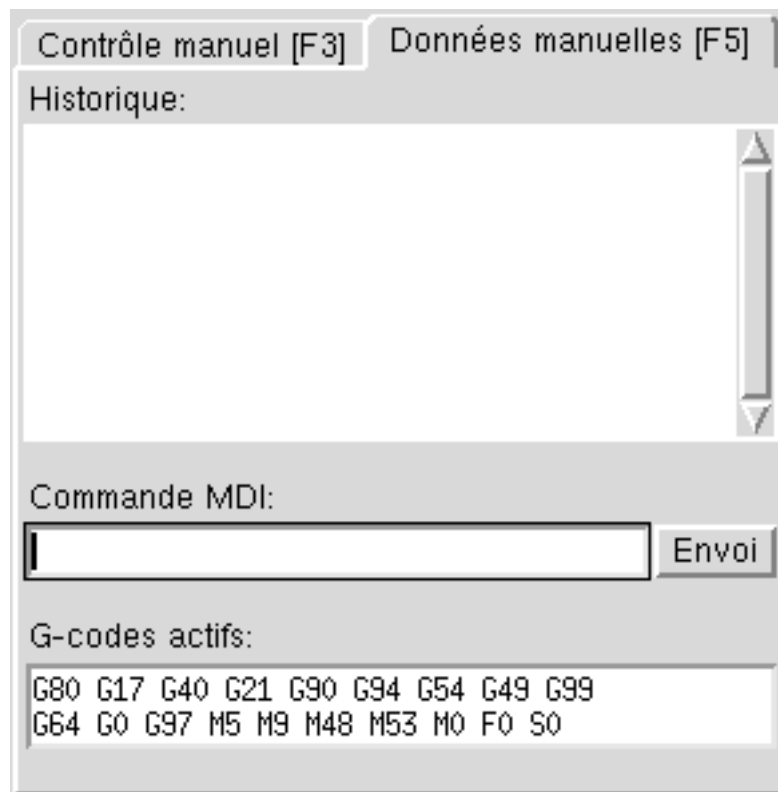


FIGURE 3.5 – L'onglet *Données manuelles*

#### **Historique**

Affiche les commandes précédemment tapées et au cours des session précédentes.

#### **Commande MDI**

Ce champ permet la saisie d'une ligne de commande à exécuter. La commande sera exécutée par l'appui de la touche <Entrée> ou un appui sur le bouton *Envoi*.

#### **G-Codes actifs**

Cette fenêtre affiche les *codes modaux* actuellement actifs dans l'interpréteur. Par exemple, **G54** indique que le système de décalage d'origine actuel est **G54** qu'il s'appliquera à toutes les coordonnées qui seront entrées.

### 3.3.7 Correcteurs de vitesse

En déplaçant le curseur, la vitesse de déplacement programmée peut être modifiée. Par exemple, si un programme requiert une vitesse à **F600** et que le curseur est placé sur 120%, alors la vitesse résultante sera de **F720**.

### 3.3.8 Correcteur de vitesse de broche

En déplaçant ce curseur, la vitesse programmée de la broche peut être modifiée. Par exemple, si un programme requiert une vitesse à F8000 et que le curseur est placé sur 80%, alors la fréquence de rotation résultante sera de **F6400**. Cet élément n'apparaît que si la *HAL pin motion.spindle-speed-out* est connectée dans le fichier .hal.

### 3.3.9 Vitesse de Jog

En déplaçant ce curseur, la vitesse de jog peut être modifiée. Par exemple, si ce curseur est placé sur 100 mm/mn, alors un jog de 1 mm durera .6 secondes, ou 1/100 de minute. Du côté gauche du curseur (jog lent) l'espacement des valeurs est petit alors que du côté droit (jog rapide) l'espacement des valeurs est plus grand, cela permet une large étendue de vitesses de jog avec un contrôle plus fin du curseur dans les zones les plus importantes.

Sur les machines avec axes rotatifs, un second curseur de vitesse est présent. Il permet d'ajuster la vitesse de rotation des axes rotatifs (A, B et C).

### 3.3.10 Vitesse Maxi

En déplaçant ce curseur, la vitesse maximale peut être réglée. Ceci couvre la vitesse maximale de tous les mouvements programmés, sauf les mouvements avec broche synchronisée.

## 3.4 Raccourcis clavier

La plupart des actions d'AXIS sont accessibles depuis le clavier. La liste complète des raccourcis clavier est disponible dans l'aide rapide d'AXIS qui s'affiche en cliquant sur Aide > Aide rapide . Beaucoup de ces raccourcis sont inaccessible en mode Entrées manuelles.

Touches des correcteurs de vitesse.

- Les touches des correcteurs de vitesse se comportent différemment en mode manuel. Les touches 12345678 sélectionneront l'axe correspondant, si il est programmé.
- Si vous avez 3 axes, alors \* sélectionnera l'axe 0, 1 sélectionnera l'axe 1, et 2 sélectionnera l'axe 2.
- Pendant l'exécution d'un programme, les touches 1234567890 fixeront la correction de vitesse travail entre 10% et 100%.

Les raccourcis clavier les plus fréquents sont visibles dans la table ci-dessous.

TABLE 3.1: Raccourcis clavier usuels

Touches	Actions produites	Mode
F1	Bascule l'arrêt d'urgence	Tous
F2	Bascule le marche/arrêt machine	Tous
`, 1 .. 9, 0	Correcteurs de vitesse de 10% à 100%	Varie
X, *	Active le premier axe	Manuel
Y, 1	Active le deuxième axe	Manuel
Z, 2	Active le troisième axe	Manuel
A, 3	Active le quatrième axe	Manuel
I	Sélection d'incrément du jog	Manuel
C	jog en mode continu	Manuel
Ctrl+origine	Lance une séquence de POM	Manuel
Fin	Toucher: valide l'offset G54 de l'axe actif	Manuel

TABLE 3.1: (continued)

Touches	Actions produites	Mode
Gauche, Droite	Jog du premier axe	Manuel
Up, Down	Jog du deuxième axe	Manuel
Pg Up, Pg Dn	Jog du troisième axe	Manuel
[, ]	Jog du quatrième axe	Manuel
O	Ouvrir un fichier	Manuel
Ctrl+R	Recharger le fichier courant	Manuel
R	Exécuter le programme	Manuel
P	Pause dans l'exécution du programme	Auto
S	Reprise de l'exécution du programme	Auto
ESC	Stopper l'exécution	Auto
Ctrl+K	Rafraichi le tracé d'outil	Auto/Manuel
V	Défilement cyclique des vues prédéfinies	Auto/Manuel
Maj-gauche, droite	Axe X vitesse rapide	Manuel
Maj-haut, bas	Axe Y vitesse rapide	Manuel
Maj-PgUp, PgDn	Axe Z vitesse rapide	Manuel

### 3.5 Affichage de l'état de LinuxCNC (LinuxCNCtop)

AXIS inclut un programme appelé *linuxcnc*top qui affiche en détail l'état de LinuxCNC. Ce programme est accessible dans le menu Machine > Fenêtre d'état de LinuxCNC.

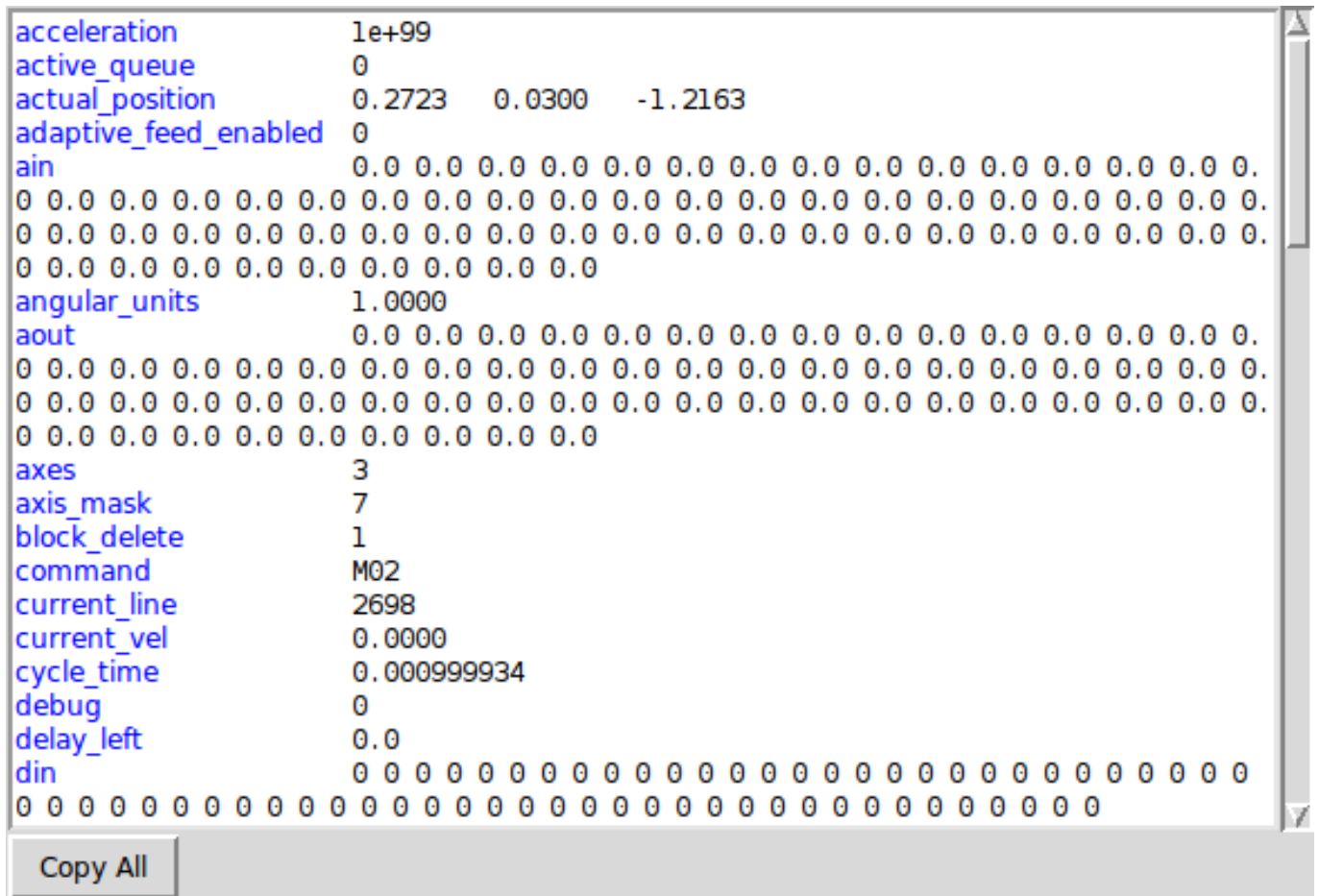


FIGURE 3.6 – Fenêtre d'état de LinuxCNC

Le nom de chaque entrée est affiché dans la colonne de gauche. La valeur courante de chaque entrée s'affiche dans la colonne de droite. Si la valeur a changé récemment, elle s'affiche en surbrillance rouge.

### 3.6 Entrée de données en texte (MDI)

AXIS inclut un programme appelé *mdi*, il permet d'envoyer des commandes à la session de LinuxCNC en cours, sous forme de lignes de texte. Vous pouvez lancer ce programme en ouvrant une console et en tapant:

```
mdi /chemin/vers/linuxcnc.nml
```

En cours d'exécution il affiche le prompt: *MDI>*. Quand une ligne vide est entrée, la position courante de la machine est affichée. Quand une commande est entrée, elle est passée à LinuxCNC qui l'exécute.

C'est un exemple de session de MDI.

```
$ MDI ~/linuxcnc/configs/sim/emc.nml
MDI>
(0.0, 0.0, 0.0, 0.0, 0.0, 0.0)
MDI> G1 F5 X1
MDI>
(0.5928500000000374, 0.0, 0.0, 0.0, 0.0, 0.0)
MDI>
(1.0000000000000639, 0.0, 0.0, 0.0, 0.0, 0.0)
```

### 3.7 axis-remote: Commande à distance de l'interface graphique d'AXIS

AXIS inclut un programme appelé *axis-remote* qui permet d'envoyer certaines commandes vers l'application AXIS fonctionnant à distance. Les commandes disponibles sont visibles en faisant: *axis-remote --help* pour vérifier qu'AXIS est en marche, inclure: (*--ping*), charger un fichier, recharger le fichier courant avec: (*--reload*) et quitter le programme AXIS avec: (*--quit*).

### 3.8 hal\_manualtoolchange: Dialogue de changement d'outil manuel

LinuxCNC inclut un composant userspace HAL de appelé *hal\_manualtoolchange*, il ouvre une fenêtre d'appel d'outil visible ci-dessous, quand la commande **M6** est invoquée. Dès que le bouton *Continuer* est pressé, l'exécution du programme reprend.

Le fichier de configuration HAL *configs/sim/axis\_manualtoolchange.hal* montre les commandes HAL nécessaires pour l'utilisation de ce composant.

*hal\_manualtoolchange* peut être utilisé même si l'interface graphique AXIS n'est pas en service. Cette composante est particulièrement utile si vous avez des outils de pré-réglage et que vous utilisez la table d'outils.



#### Important

Le parcours d'outil d'un mouvement en vitesse rapide ne sera pas visible s'il suit un changement d'outil T<n> et ce jusqu'au prochain mouvement en vitesse travail. Ceci peut être source de confusion pour l'opérateur. Pour contourner cette particularité, ajoutez toujours un G1 sans mouvement après un M6 T<n>.

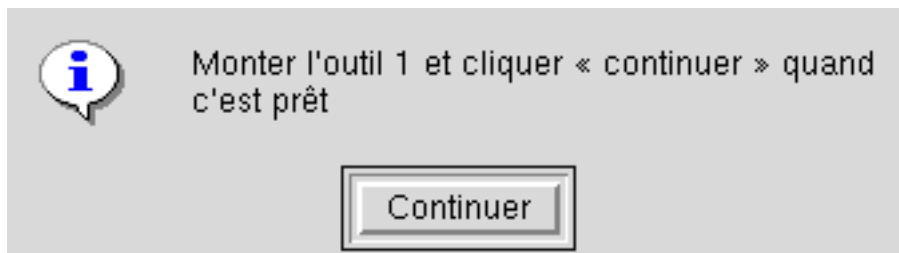


FIGURE 3.7 – La fenêtre de changement manuel d'outil

### 3.9 Modules en Python

AXIS inclut plusieurs modules en Python qui peuvent être très utiles. Pour des informations complètes sur ces modules, faites: *pydoc <nom du module>* ou lisez son code source. Modules inclus:

- **LinuxCNC** fournit l'accès aux commandes de LinuxCNC, à son état et aux chaînes d'erreur
- **gcode** fournit l'accès à l'interpréteur RS274NGC
- **rs274** fournit des outils supplémentaires pour travailler sur les fichiers RS274NGC
- **hal** permet la création par l'utilisateur de composants de HAL écrits en Python
- **\_togl** fournit des éléments OpenGL utilisables dans les applications Tkinter
- **minigl** fournit l'accès aux sous-ensembles d'OpenGL utilisés par AXIS

Pour utiliser ces modules dans vos propres scripts, assurez-vous que le répertoire où ils se trouvent est dans le chemin d'accès des modules Python. Avec une version installée de LinuxCNC, ça se fera automatiquement. Avec une version installée en *in-place*, ça peut être fait avec l'aide de: */scripts/rip-environment*.

### 3.10 Utiliser AXIS en mode tour

En incluant la ligne

LATHE = 1

dans la section [DISPLAY] du fichier ini, AXIS sélectionnera le mode tour. L'axe Y ne s'affiche pas parmi les coordonnées, la vue est modifiée pour placer la broche à gauche, l'axe Z horizontalement avec son côté positif vers la droite (**Z+**) et l'axe X s'étendant vers le bas de l'écran (**X+**). Plusieurs contrôles (tels que ceux des vues prédéfinies) sont supprimés. Les lectures de coordonnées pour X sont désormais en diamètre et en rayon.

La touche **V** agit alors sur le zoom pour afficher le tracé complet du fichier chargé.

En mode tour (lathe), la forme et l'orientation de l'outil chargé sont représentés.

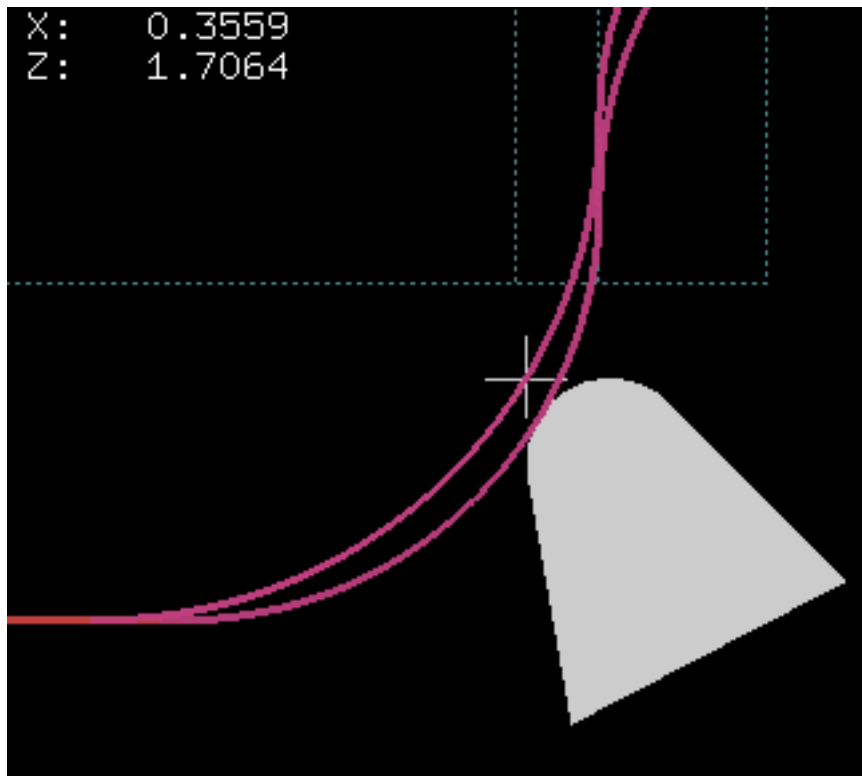


FIGURE 3.8 – Représentation de l'outil en mode tour

### 3.11 Configuration avancée d'AXIS

Pour plus d'informations sur les paramètres du fichier ini pouvant modifier le fonctionnement d'AXIS, voir le fichier ini, sections [DISPLAY] et le chapitre sur la configuration dans le manuel de l'intégrateur.

#### 3.11.1 Filtres de programme

AXIS a la capacité d'envoyer des fichiers chargés à travers un *filtre de programme*. Ce filtre peut faire diverses tâches: Simple, comme s'assurer que le programme se termine bien par un **M2** ou complexe, comme détecter que l'entrée est une image et générer le g-code qui permettra d'usiner sa forme.

La section [FILTER] du fichier ini définit comment les filtres doivent agir. Premièrement, pour chaque type de fichier, écrire une ligne PROGRAM\_EXTENSION puis, spécifier le programme à exécuter pour chaque type de fichier. Ce programme reçoit comme argument le nom du fichier d'entrée, il doit produire le G-code selon le standard rs274ngc, en sortie. Les lignes de cette sortie s'affichent alors dans la zone de texte, le parcours d'outil résultant est visible dans la zone graphique, enfin il sera exécuté quand LinuxCNC recevra la commande *Exécuter le programme*. Les lignes suivantes fournissent la possibilité d'utiliser *image-to-gcode*, le convertisseur d'images fourni avec LinuxCNC:

```
[FILTER]
PROGRAM_EXTENSION = .png,.gif Greyscale Depth Image
png = image-to-gcode
gif = image-to-gcode
```

Il est également possible de spécifier un interpréteur:

```
PROGRAM_EXTENSION = .py Python Script
py = python
```

De cette manière, n'importe quel script Python pourra être ouvert et sa sortie traitée comme du G-code. Un autre exemple est disponible dans: `/nc_files/holecircle.py`. Ce script crée le G-code pour percer une série de trous suivant un arc de cercle.

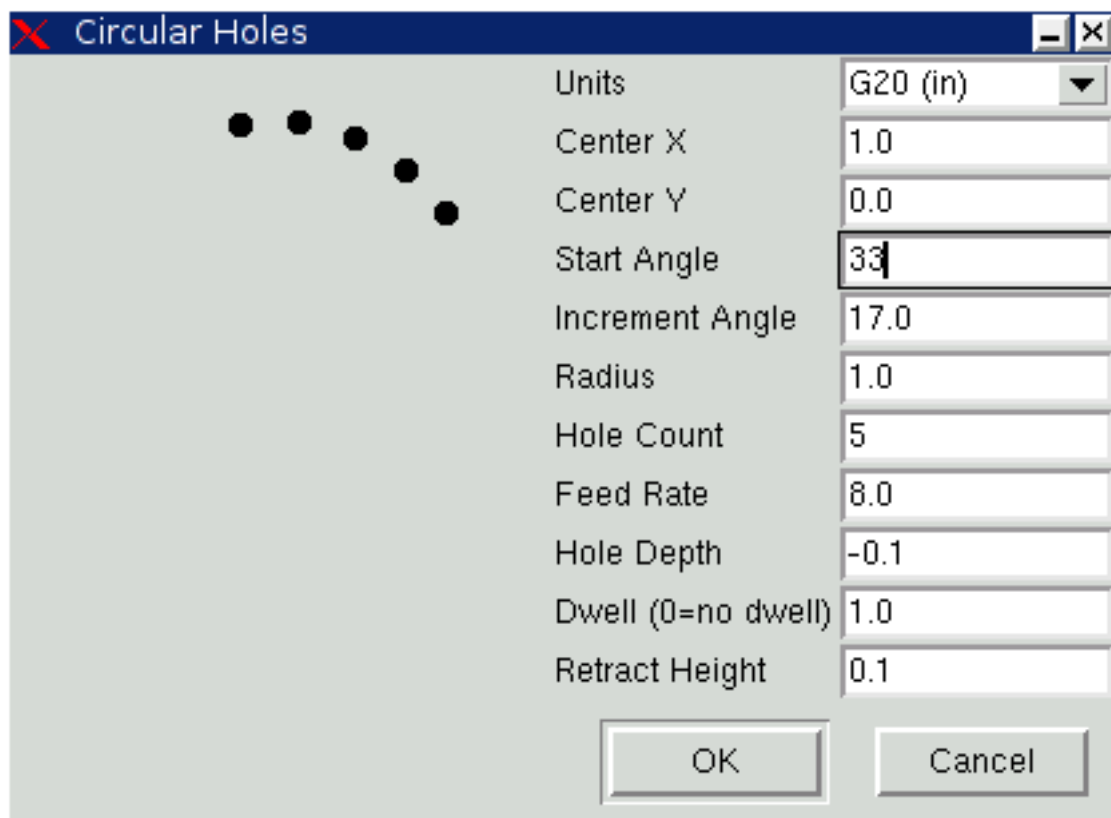


FIGURE 3.9 – Perçages circulaires

Si la variable d'environnement: `AXIS_PROGRESS_BAR` est active, alors les lignes seront écrites sur stderr de la forme:

```
FILTER_PROGRESS=%d
```

AXIS fixera la barre de progression selon le pourcentage donné. Cette fonction devrait être utilisée pour un filtre qui fonctionne suffisamment longtemps.

### 3.11.2 La base de données des ressources X

Les couleurs de la plupart des éléments de l'interface utilisateur d'AXIS peuvent être personnalisées grâce à la base de données X. Le fichier `axis_light_background` modifie les couleurs de la fenêtre du parcours d'outil sur le modèle *lignes noires et fond blanc*, il sert aussi de référence des éléments configurables dans l'écran graphique. L'exemple de fichier `axis_big_dro` évolution de la position de lecture à une police plus grande taille. Pour utiliser ces fichiers:

```
xrdb -merge /usr/share/doc/linuxcnc/axis_light_background
xrdb -merge /usr/share/doc/linuxcnc/axis_big_dro
```

Pour plus d'informations au sujet des éléments configurables dans les applications Tk, référez vous aux manuels de Tk.

Les bureaux graphiques modernes effectuent certains réglages dans la base de données des ressources X ces réglages peuvent affecter ceux d'AXIS, par défaut ces réglages sont ignorés. Pour que les éléments des ressources X écrasent ceux par défaut dans AXIS, il faut inclure cette ligne dans vos ressources X:

```
*Axis*optionLevel: widgetDefault
```

ce qui entraînera la construction des options au niveau *widgetDefault*, de sorte que les ressources X (qui sont elles, au niveau *userDefault*) puissent l'emporter.

### 3.11.3 Manivelle de jog

Pour accroître l'interaction d'AXIS avec une manivelle de jog physique, l'axe actif courant sélectionné dans l'interface graphique est aussi reporté sur une *pin HAL* avec un nom comme *axisui.jog.x*. Excepté pendant un court instant après que l'axe courant ait changé, une seule de ces pins à la fois est *TRUE*, les autres restent *FALSE*.

Après qu'AXIS ait créé ces *HAL pins*, il exécute le fichier hal déclaré avec: [HAL]POSTGUI\_HALFILE. Ce qui diffère de [HAL]HALFILE, qui lui ne s'utilise qu'une seule fois.

### 3.11.4 ~/.axisrc

Si il existe, le contenu de: *~/.axisrc* est exécuté comme un code source Python juste avant l'ouverture de l'interface graphique d'AXIS. Les détails de ce qui peut être écrit dans *.axisrc* sont sujets à changement durant le cycle de développement.

Les lignes visibles ci-dessous ajoutent un Ctrl+Q comme raccourci clavier pour Quitter et activer l'option *Distance restante* par défaut.

#### Exemple de fichier .axisrc

```
root_window.bind("<Control-q>", "destroy .")
help2.append(("Control-Q", "Quit"))
```

### 3.11.5 Éditeur externe

En définissant: [DISPLAY]EDITOR , les options de menu: *Fichier* → *Éditer* ainsi que *Fichier* → *Éditer la table d'outils*, deviennent accessibles. Deux valeurs qui fonctionnent bien: EDITOR=gedit et EDITOR=gnome-terminal -e nano.

### 3.11.6 Panneau de contrôle virtuel

AXIS peut afficher un panneau de commande virtuel personnalisé dans le volet de droite. Il est possible d'y placer des boutons, des indicateurs qui afficheront des données et plus encore. Voir le manuel de l'intégrateur.

### 3.11.7 Commentaires spéciaux

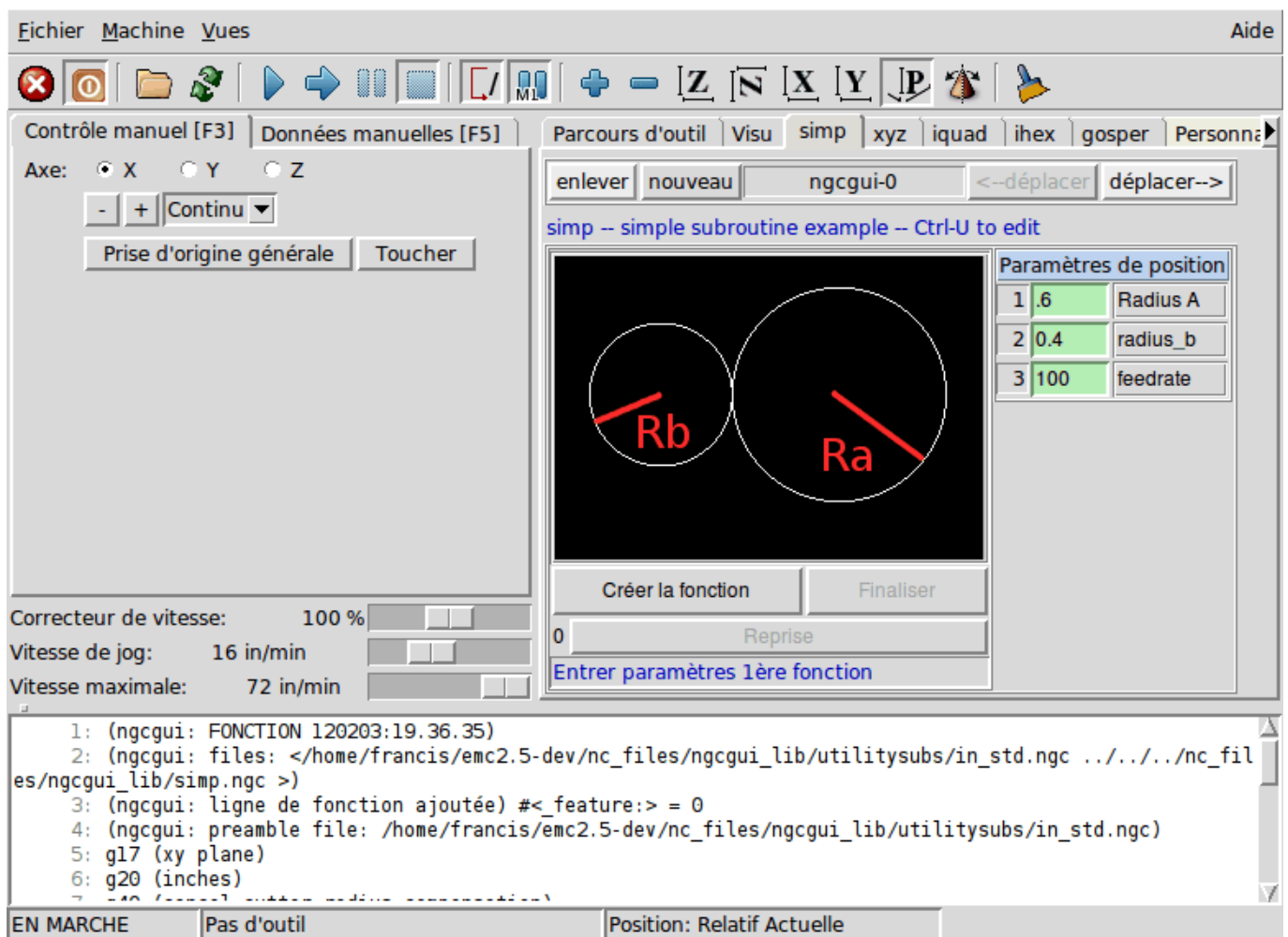
Les commentaires spéciaux peuvent être insérés dans le fichier de G-code pour contrôler le comportement de l'affichage d'AXIS. Pour limiter l'aperçu au seul affichage du parcours d'outil, utiliser ces commentaires spéciaux. Rien ne s'affichera entre le commentaire (AXIS,hide) et le commentaire (AXIS,show) sauf le parcours d'outil. Les (AXIS,hide) et (AXIS,show) doivent être utilisés par paires avec le (AXIS, hide) en premier. Tout ce qui est après un (AXIS,stop) ne sera pas visible.

Ces commentaires sont utiles pour désencombrer l'affichage d'aperçu (Par exemple pendant le débogage d'un grand fichier G-code, on peut désactiver l'aperçu sur certaines parties qui sont déjà fonctionnelles).

- (AXIS,hide) Arrête le parcours d'outil (à placer en premier)
- (AXIS,show) Reprend le parcours d'outil (il faut suivre un cache)
- (AXIS,stop) Arrête le parcours d'outil d'ici à la fin du fichier.
- (AXIS,notify,le\_texte) Affiche *le\_texte* à l'écran, comme une info. Cet affichage peut être très utile lors du pré-affichage du parcours d'outil, quand les commentaires (debug,message) ne sont pas affichés.

## Chapitre 4

# L'utilitaire graphique NGCGUI



### 4.1 Vue d'ensemble

- NGCGUI est un utilitaire pour écrire et utiliser les sous-programmes avec LinuxCNC.
- NGCGUI peut être utilisé en autonome ou intégré, dans ce dernier cas, il crée des onglets multiples sur la page de l'interface graphique Axis.




Ngcgui est un outil puissant pour construire les programmes de G-code en sous-programmes.

- Les sous-programmes peuvent être concaténés pour fournir un programme de G-code complet.
- De multiples instances d'un sous-programme peuvent être utilisées pour fournir la même tâche à différents emplacements sur la même pièce.
- N'importe quel G-code valide peut être utilisé dans un sous-programme.
- De nouveaux sous-programmes peuvent être ajoutés à la volée.

## 4.2 Configurations fournies en exemple.

Trois configurations sont fournies, elles se trouvent dans le répertoire *sim* du sélecteur de configuration de LinuxCNC. Le sélecteur de configuration se trouve quand à lui dans le menu *Applications* → *CNC* → *LinuxCNC*.

- *ngcgui* - Un exemple facile à comprendre utilisant ces sous-programmes:
  - *simp* - Un exemple simple créant deux cercles.
  - *xyz* - Crée une boîte basée sur deux coins opposés.
  - *iquad* - Crée un quadrilatère interne.
  - *db25* - Crée la découpe pour une fiche DB25.
  - *ihex* - Crée un hexagone interne.
  - *gosper* - Une démo sur la récursion.
  - *Custom* - Crée des onglets personnalisés.
  - *ttt* - Traceur True Type, pour créer des textes à graver.
- *ngcgui-lathe* - Un exemple de sous-programme pour un tour:
  - *id* - Alésage intérieur.
  - *od* - Cylindrage extérieur.
  - *taper-od* - Tourne un cône mâle.
  - *Custom* - Crée des onglets personnalisés.
- *ngcgui-simple* - Un exemple simple:
  - *simp* - Un exemple simple créant deux cercles.
  - *xyz* - Crée une boîte basée sur deux coins opposés.

Pour visualiser les sous-programmes presser l'**Arrêt d'Urgence**  puis, activer la **Marche Machine**  et réaliser la **Prise d'origine générale**. Cliquer sur un onglet de *ngcgui* et presser **Créer la fonction** puis **Finaliser**. Enfin, presser sur le bouton  **Départ cycle** pour exécuter le G-code.

### Note

Les sous-programmes d'exemples fournis avec la distribution doivent tous fonctionner avec la configuration de la machine simulée. Un utilisateur doit toujours comprendre le comportement et les implications d'un programme avant de tenter de l'exécuter sur une machine réelle.

## 4.3 Librairies

Les configurations en simulation pour *ngcgui* utilisent les liens suivants vers des librairies de LinuxCNC protégées en écriture:

- *Sous-fichiers compatibles ngcgui* - *ngcgui\_lib*
- *Sous-programme d'aide* - *ngcgui\_lib/utilitysubs*
- *Fichiers M utilisateurs* - *ngcgui\_lib/mfiles*

Ces librairies sont définies dans le fichier ini par les items:

```
[RS274NGC]
SUBROUTINE_PATH = ../../../../nc_files/ngcgui_lib:../../../../nc_files/ngcgui_lib/utilitysubs
USER_M_PATH      = ../../../../nc_files/ngcgui_lib/mfiles
```

C'est une longue ligne (ne pas continuer sur de multiples lignes) qui spécifie les répertoires utilisés dans le chemin de recherche. Les noms de répertoires sont séparés par le caractère (:).

L'utilisateur peut créer de nouveaux répertoires pour ses propres sous-programmes et fichiers M et les ajouter dans le chemin de recherche.

Par exemple, un utilisateur pourrait créer ces répertoires à partir de la console.

```
mkdir /home/myusername/mysubs  
mkdir /home/myusername/mymfiles
```

Puis y créer ou y copier des fichiers qui lui seront accessibles en écriture. Par exemple, créer un sous-fichier compatible ngcgui nommé:

```
/home/myusername/mysubs/exemple.ngc
```

Le fichier ini doit être édité pour lui inclure les nouveaux sous-fichiers et les ajouter au chemin. Pour cet exemple:

```
[RS274NGC]  
SUBROUTINE_PATH = /home/myusername/mysubs:../../../../nc_files/ngcgui_lib:../../../../nc_files/ ↔  
                  ngcgui_lib/utilitysubs  
USER_M_PATH      = /home/myusername/mymfiles:../../../../nc_files/ngcgui_lib/mfiles  
  
[DISPLAY]  
NGCGUI_SUBFILE = exemple.ngc
```

LinuxCNC et ngcgui utilisent le premier fichier trouvé lors d'une recherche dans les répertoires du chemin de recherche. Avec ce comportement, Il est possible de substituer un sous-fichier ngcgui\_lib en plaçant un sous-fichier avec un nom identique plus tôt dans le chemin de recherche pour qu'il soit trouvé avant. Plus d'informations peuvent être trouvées au chapitre INI dans le Manuel de l'intégrateur.

## 4.4 Intégration de ngcgui dans Axis

D'autres exemples de sous-programmes se trouvent dans le répertoire sim/ngcgui

Les items de fichier INI pour NGCGUI vont dans la section [DISPLAY].

- *TKPKG* = Ngcgui 1.0 - Le paquet principal de NGCGUI (doit précéder Ngcguittt).
- *TKPKG* = Ngcguittt 1.0 - Le paquet du traceur True Type pour générer des textes à graver.
- *NGCGUI\_FONT* = Helvetica -12 normal - Spécifie la police utilisée.
- *NGCGUI\_PREAMBLE* = in\_std.ngc - Le fichier de préambule à ajouter au début du sous-programme. Quand plusieurs sous-programmes sont concaténés, un seul est ajouté.
- *NGCGUI\_SUBFILE* = simp.ngc - Crée un onglet depuis le sous-programme nommé.
- *NGCGUI\_SUBFILE* = "" - Crée un onglet personnalisé.
- *#NGCGUI\_OPTIONS* = opt1 opt2 ... - Options Ngcgui
  - # opt items:
    - # nonew — interdit la création d'un nouvel onglet personnalisé
    - # noremove — interdit l'effacement d'une page d'onglet
    - # noauto — pas d'envoi auto (makeFile, puis envoi manuel)
    - # noiframe — no internal image, image on separate top level
- *TTT* = Le programme True-type Tracer
- *TTT\_PREAMBLE* = in\_std.ngc - Optionnel, spécifie le nom de fichier de préambule utilisé par ttt pour créer les sous-fichiers.

Voici un exemple d'intégration de NGCGUI dans Axis. Les sous-programmes doivent être placés dans un répertoire spécifié par la variable [RS274NGC]SUBROUTINE\_PATH. Certains exemples de sous-programmes utilisent d'autres sous-programmes, bien vérifier pour être sûr d'avoir les bonnes dépendances, le cas échéant, dans un répertoire SUBROUTINE\_PATH. Certains sous-programmes peuvent utiliser des fichiers M (Mfiles) personnalisés qui doivent se trouver dans un répertoire spécifié par [RS274NGC]USER\_M\_PATH.

---

### Note

Il ne s'agit pas d'un fichier ini complet, les items montrés sont ceux utilisés par ngcgui. D'autres items sont requis par LinuxCNC pour obtenir un fichier ini complet.

---

### Simple fichier.ini

```
[RS274NGC]
SUBROUTINE_PATH  = ../../../../nc_files/ngcgui_lib:../../../../ngcgui_lib/utilitysubs
USER_M_PATH      = ../../../../nc_files/ngcgui_lib/mfiles

[DISPLAY]
TKPKG           = Ngcgui      1.0
TKPKG           = Ngcgui_ttt 1.0
# Ngcgui must precede Ngcgui_ttt

NGCGUI_FONT      = Helvetica -12 normal
# specifie seulement les noms de fichiers, doit être dans [RS274NGC]SUBROUTINE_PATH
NGCGUI_PREAMBLE  = in_std.ngc
NGCGUI_SUBFILE   = simp.ngc
NGCGUI_SUBFILE   = xyz.ngc
NGCGUI_SUBFILE   = iquad.ngc
NGCGUI_SUBFILE   = db25.ngc
NGCGUI_SUBFILE   = ihex.ngc
NGCGUI_SUBFILE   = gosper.ngc
# specifie "" pour une page d'onglet épersonnalise
NGCGUI_SUBFILE   = ""
#NGCGUI_SUBFILE  = "" éutilis quand une trame d'image est ééspecifie si
#                  ouvrir d'autres fichiers est requis
#                  les images seront mises dans une éfentre de haut niveau
NGCGUI_OPTIONS   =
#NGCGUI_OPTIONS  = opt1 opt2 ...
# opt items:
# new            -- interdit la écration d'un nouvel onglet épersonnalis
# noremove       -- interdit l'effacement d'une page d'onglet
# noauto         -- pas d'envoi auto (makeFile, puis envoi manuel)
# noiframe       -- no internal image, image on separate top level

TTT              = truetype-tracer
TTT_PREAMBLE     = in_std.ngc

PROGRAM_PREFIX    = ../../nc_files
```

#### 4.4.1 Traceur Truetype

Ngcgui\_ttt fourni le support pour truetype-tracer (v4). Il crée un onglet sur Axis qui permet à l'utilisateur de créer ses propres textes dans de nouveaux onglets ngcgui et en choisissant leurs fontes et autres paramètres. (Truetype-tracer doit être installé indépendamment).

L'intégration de ngcgui\_ttt dans Axis, nécessite les items suivants en plus de ceux de ngcgui:

Item: [DISPLAY]TKPKG = Ngcgui\_ttt numéro\_de\_version  
Exemple: [DISPLAY]TKPKG = Ngcgui\_ttt 1.0  
Note: Obligatoire, spécifie le chargement de ngcgui\_ttt dans un onglet d'Axis nommé ttt. Doit suivre l'item TKPKG = Ngcgui.

Item: [DISPLAY]TTT = chemin\_de\_truetype-tracer  
Exemple: [DISPLAY]TTT = truetype-tracer  
Note: Optionnel, s'il n'est pas spécifié, utilisera /usr/local/bin/truetype-tracer. Spécifier avec un chemin absolu ou simplement le nom de l'exécutable, dans ce cas, la variable d'environnement PATH de l'utilisateur sera utilisée pour rechercher le programme.

Item: [DISPLAY]TTT\_PREAMBLE = nom\_fichier\_préambule

Exemple: [DISPLAY]TTT\_PREAMBLE = in\_std.ngc

Note: Optionnel, spécifie le nom du fichier de préambule utilisé pour les sous-fichiers créés par ttt.

#### 4.4.2 Exemples d'INI

Ngcgui utilise le chemin de recherche de LinuxCNC pour chercher les fichiers.

Le chemin de recherche commence avec le répertoire standard spécifié par:

```
[DISPLAY]PROGRAM_PREFIX
```

suivi par les répertoires multiples spécifiés par:

```
[RS274NGC]SUBROUTINE_PATH
```

**Répertoires** Les répertoires peuvent être spécifiés comme des chemins absolus ou des chemins relatifs.

Exemple: [DISPLAY]PROGRAM\_PREFIX = /home/myname/linuxcnc/nc\_files

Exemple: [DISPLAY]PROGRAM\_PREFIX = ~/linuxcnc/nc\_files

Exemple: [DISPLAY]PROGRAM\_PREFIX = ../../../nc\_files

**Chemins absolus** Un chemin absolu commence avec un "/" qui indique un emplacement par rapport au système de fichiers complet. Un chemin qui commence par "~/" indique un chemin commençant *depuis* le répertoire home de l'utilisateur. Un chemin qui commence par "~nomutilisateur/" indique un chemin commençant *dans* le répertoire utilisateur.

**Chemins relatifs** Un chemin relatif commence dans le répertoire de démarrage qui est celui contenant le fichier ini. L'usage des chemins relatifs facilite l'accès aux configurations mais requiert une bonne compréhension de la façon dont les chemins sont spécifiés sous Linux.

```
./d0      est le même que d0, ex: un répertoire nommé d0 dans le répertoire de
départ.
../d1     se réfère au répertoire d1 dans le répertoire parent.
../../d2  se réfère au répertoire d2 dans le répertoire parent du parent.
../../../d3 etc.
```

Des répertoires multiples peuvent être spécifiés par la variable: [RS274NGC]SUBROUTINE\_PATH suivie des chemins séparés par le signe ":". L'exemple suivant illustre le format utilisé pour les chemins multiples et montre l'utilisation de répertoires relatifs et absolus.

Exemple: [RS274NGC]SUBROUTINE\_PATH = ../../../nc\_files/ngcgui\_lib: ../../../nc\_files/ngcgui\_lib/utilitysubs:/tmp/tmpngc

C'est une longue ligne, ne pas continuer sur de multiples lignes. Quand LinuxCNC et/ou Ngcgui cherchent un fichier, c'est le premier trouvé qui est utilisé.

LinuxCNC (et NGCGUI) doivent pouvoir trouver tous les sous-programmes avec les routines additionnelles qui sont appelées depuis les sous-fichiers NGCGUI. Il est pratique de placer les fichiers utilitaires dans un répertoire séparé comme indiqué dans l'exemple précédent.

La distribution inclut le répertoire ngcgui\_lib et les fichiers de préambule, sous-fichiers, postambule et fichiers d'aide pour les démos. Pour modifier le comportement des fichiers, il est possible de copier n'importe quel fichier et de le placer en avant du chemin de recherche. Le premier répertoire recherché est: [DISPLAY]PROGRAM\_PREFIX. Il est possible de l'utiliser mais c'est une meilleure pratique de créer un répertoire dédié en le plaçant au début du chemin donné par [RS274NGC]SUBROUTINE\_PATH.

Dans l'exemple suivant, les fichiers dans /home/myname/emc2/mysubs seront trouvés avant ceux étant dans ../../../nc\_files/ngcgui\_lib.

Exemple: [RS274NGC]SUBROUTINE\_PATH = /home/myname/emc2/mysubs: ../../../nc\_files/ngcgui\_lib: ../../../nc\_files/ngcgui\_lib/utility

Les débutants pourraient essayer par inadvertance d'utiliser des fichiers non structurés comme le nécessite ngcgui. Ngcgui déclencherait alors rapidement de nombreuses erreurs si les fichiers ne répondent pas à ses conventions. Une bonne pratique suggère que les sous-fichiers compatibles ngcgui doivent être placés dans un répertoire dédié à cette fin et que les préambules, postambules et fichiers d'aide doivent être dans un répertoire séparés pour dissuader toute tentative d'utilisation de ces sous-fichiers.

Pour intégrer ngcgui dans Axis, spécifier les items suivants dans le fichier ini:

Item: [DISPLAY]PROGRAM\_PREFIX = dirname

Exemple: [DISPLAY]PROGRAM\_PREFIX = ../../../../nc\_files

Note: Obligatoire et nécessaire pour de nombreuses fonctions de LinuxCNC.  
C'est le premier répertoire utilisé lors de la recherche de fichiers.

Item: [RS274NGC]SUBROUTINE\_PATH = dirname1:dirname2:dirname3 ...

Exemple: [RS274NGC]SUBROUTINE\_PATH = ../../../../nc\_files/ngcgui\_lib:../../../../nc\_files/ngcgui

Note: Optionnel, mais très utile pour organiser les sous-fichiers et les fichiers utilitaires.

Item: [DISPLAY]TKPKG=Ngcgui version\_number

Exemple: [DISPLAY]TKPKG=Ngcgui 1.0

Note: Obligatoire, spécifie le chargement des onglets ngcgui dans axis.

Item: [DISPLAY]NGCGUI\_FONT = font\_descriptor

Exemple: [DISPLAY]NGCGUI\_FONT = Helvetica -12 normal

Note: Optionnel, descripteur de fontes compatible avec celui de Tcl.  
Avec des items pour le type de fonte -fontsize fontweight  
Par défaut c'est la police: Helvetica -10 normal

Item: [DISPLAY]NGCGUI\_SUBFILE = subfile\_filename

Exemple: [DISPLAY]NGCGUI\_SUBFILE = simp.ngc

Exemple: [DISPLAY]NGCGUI\_SUBFILE = xyz.ngc

Exemple: [DISPLAY]NGCGUI\_SUBFILE = ""

Note: Utilise un ou plusieurs items pour spécifier les fichiers compatibles avec les sous-fichiers ngcgui qui requiert un onglet dans Axis au départ. Un onglet "personnalisé" est créé quand le nom de fichier est "". Un utilisateur peut utiliser l'onglet "Personnalisé" pour lire un fichier système et identifier un préambule, un sous-fichier ou un postambule.

Item: [DISPLAY]NGCGUI\_PREAMBLE = preamble\_filename

Exemple: [DISPLAY]NGCGUI\_PREAMBLE = in\_std.ngc

Note: Optionnel, si spécifié, alors ce fichier sera prépondérant à tous les sous-fichiers. Les fichiers créés avec l'onglet "Personnalisé" utilisent le préambule spécifié avec cette page.

Item: [DISPLAY]NGCGUI\_POSTAMBLE = postamble\_filename

Exemple: [DISPLAY]NGCGUI\_POSTAMBLE = bye.ngc

Note: Optionnel, si spécifié, le fichier est ajouté à tous les sous-fichiers. Les fichiers créés avec l'onglet "Personnalisé" utilisent le postambule spécifié avec cette page.

Item: [DISPLAY]NGCGUI\_OPTIONS = opt1 opt2 ...

Exemple: [DISPLAY]NGCGUI\_OPTIONS = neww noremove

Note: De multiples options séparées par des blancs.  
Par défaut, ngcgui gère les onglets de cette manière:  
1) Un utilisateur peut créer de nouveaux onglets.  
2) Un utilisateur peut enlever des onglets (excepté le dernier restant)

- 3) La finalisation envoie automatiquement les fichiers à Axis.
- 4) Une trame d'image (iframe) est rendue disponible pour afficher une image pour le sous-fichier.

Les options `_nonew_`, `_noremove_`, `_noauto_`, `_noiframe_` respectivement, désactivent ces comportements par défaut.

Par défaut, Si un fichier d'image (.png, .gif, .jpg, .pgm) est trouvé dans le même répertoire que le sous-fichier, l'image est affichée dans une iframe. Spécifier l'option `_noiframe_` rendra disponibles d'autres boutons pour sélectionner un préambule, un sous-fichier ou un postambule et des cases à cocher additionnelles. Les cases à cocher sont toujours disponibles avec les touches spéciales suivantes:  
Ctrl-R Bascule "Conserver les valeurs à la lecture du sous-fichier"  
Ctrl-E Bascule "Déployer le sous-programme"  
Ctrl-a Bascule "EnvoiAuto"  
(Ctrl-k lists all keys and functions)

Si `_noiframe_` est spécifié et qu'un fichier image est trouvé, l'image est affichée dans une fenêtre séparée et toutes les fonctions sont disponibles dans la page de l'onglet.

Les `_NGCGUI_OPTIONS_` s'appliquent à tous les onglets ngcgui excepté ceux sur lesquels les options `_nonew_`, `_noremove_`, et `_noiframe_` ne sont pas applicables pour l'onglet "Personnalisé".  
Ne pas utiliser l'onglet "Personnalisé" si les utilisateurs doivent avoir des possibilités de sélection de sous-fichiers et de création d'onglet additionnels limitées.



## 4.5 Besoins des sous-programmes

Un sous-fichiers compatible NGCGUI contient une simple définition de sous-programme. Le nom du sous-programme doit être le même que celui du fichier (non inclus l'extension .ngc). LinuxCNC supporte les sous-programmes nommés ou numérotés, mais seuls les sous-programmes nommés sont compatible avec NGCGUI. Pour plus d'informations voir le chapitre sur les [O-Codes](#).

La première ligne, autre qu'un commentaire, doit être une déclaration *sub*. La dernière ligne, autre qu'un commentaire, doit être une déclaration *endsub*.

### exemple.ngc:

```
o<exemple> sub
  CORPS DU SOUS-PROGRAMME
o<exemple> endsub
```

Le corps du sous-programme doit commencer par un jeu de déclarations définissant les paramètres nommés locaux pour chaque paramètre positionnel attendu pour l'appel du sous-programme. Ces définitions doivent être consécutives, commencer par #1 et finir avec le numéro du dernier paramètre utilisé. Les définitions doivent être fournies pour chacun de ces paramètres (aucune omissions).

### Numérotation des paramètres

```
#<xparm> = #1
#<yparm> = #2
#<zparm> = #3
```

LinuxCNC considère tous les paramètres numérotés entre #1 et #30 comme étant des paramètres appelables, de même, ngcgui fourni des dialogues de saisie pour n'importe quel paramètres dans cette fourchette. Il est de bonne pratique d'éviter d'utiliser

un paramètre numéroté de #1 jusqu'à #30 n'importe où ailleurs dans le sous-programme. L'utilisation de paramètres nommés locaux est recommandée pour toutes les variables internes.

Chaque définition de déclaration peut optionnellement inclure un commentaire spécial et une valeur par défaut pour le paramètre.

### Prototypage de déclaration

```
#<vname> = #n (=évalueur_par_dfauf)
ou
#<vname> = #n (texte_de_commentaire)
ou
#<vname> = #n (=évalueur_par_dfauf texte_de_commentaire)
```

### Exemples de paramètres

```
#<xparm> = #1 (=0.0)
#<yparm> = #2 (Ystart)
#<zparm> = #3 (=0.0 Z start setting)
```

Si une valeur\_par\_défaut est donnée, elle sera placée au démarrage, dans la boîte de saisie pour le paramètre.

Si un texte\_de\_commentaire est inclus, il sera utilisé pour identifier l'entrée à la place du nom du paramètre.

**Paramètres nommés globaux** Note sur les paramètres nommés globaux (#<nom\_global>) avec ngcgui:

Comme dans de nombreux langages de programmation, l'utilisation de variables globales est puissante, mais peut souvent mener à des conséquences inattendues. Dans LinuxCNC, les paramètres nommés globaux existants seront valides lors de l'exécution du sous-programme et les sous-programmes peuvent les modifier ou en créer.

L'utilisation de paramètres nommé globaux comme entrées dans un sous-programme est déconseillé parce-que de tels usages requiert l'établissement et la maintenance d'un contexte global bien défini, ce qui est problématique à maintenir. L'utilisation de paramètres numérotés en #1 et #30 devrait être suffisant pour satisfaire les besoins les plus exigeants.

Ngcgui supporte quelques entrées par paramètres nommés globaux mais leurs usage est obsolète et non documenté ici.

Bien que les entrées par paramètres nommés globaux soient déconseillées, les sous-programmes LinuxCNC doivent utiliser des paramètres nommés globaux pour retourner les résultats. Puisque les sous-fichiers compatibles ngcgui sont destinés à l'usage de l'interface graphique, les valeurs de retour n'ont pas d'exigence commune. Toutefois, ngcgui est utile comme outil de test pour les sous-programmes qui ne retournent pas de paramètres nommés globaux et il est commun pour les sous-fichiers compatibles ngcgui d'appeler des fichiers de sous-programmes utilitaires qui eux retournent des résultats avec des paramètres nommés globaux.

Pour supporter ces usages, ngcgui ignore les paramètres nommés globaux qui incluent le caractère (:) dans leur nom. Utilisation des deux points (:) dans un nom prévient ngcgui de créer une boîte de saisie pour ces paramètres.

### Paramètres nommés globaux

```
o<exemp> sub
...
#<_exemp:result> = #5410      (retourne le diamètre de l'outil courant)
...
o<helper> call [#<x1>] [#<x2>] (appel d'un sous-programme)
#<xresult> = #<_helper:answer> (localise immédiatement le résultat du
fichier d'aide)
#<_helper:answer> = 0.0 (rend nul le paramètre nommé global utilisé par le
sous-programme)
...
o<exemp> endsub
```

Dans l'exemple précédent, le sous-programme utilitaire sera trouvé dans un fichier séparé nommé helper.ngc. Le sous-programme d'aide retourne un résultat dans un paramètre nommé global nommé #<\_helper:answer>.

Pour une bonne pratique, le sous-fichier appelant localise immédiatement le résultat pour une utilisation ailleurs dans le sous-fichier et le paramètre nommé global, utilisé pour retourner le résultat est mis à zéro pour diminuer les chances qu'il soit utilisé par inadvertance ailleurs dans le contexte global. (La mise à zéro avec 0.0 n'est pas toujours le meilleur choix).

Ngcgui supporte la création et la concaténation de multiples fonctions pour un sous-fichier et pour de multiples sous-fichiers. Il est parfois pratique pour les sous-fichiers de déterminer leur ordre au début de l'exécution afin que ngcgui insère un paramètre global spécial qui pourra être testé par tous les sous-programmes. Ce paramètre est nommé #<\_feature:>. Sa valeur commence avec 0 et est incrémentée avec chaque fonction qui lui est ajoutée.

**Fonctions additionnelles** Un commentaire spécial *info* peut être inclus quelque part dans les sous-fichier compatibles ngcgui. Le format est le suivant:

```
(info: info_text)
```

La chaîne *info\_text* est affichée vers le haut de la page de l'onglet ngcgui dans Axis.

Les fichiers non destinés à servir de sous-fichiers peuvent inclure le commentaire spécial: "(not\_a\_subfile)" de sorte que ngcgui les rejette automatiquement avec un message explicatif.

```
(not_a_subfile)
```

Un fichier image optionnel (.png, .gif, .jpg, .pgm) peut accompagner un sous-fichier. Le fichier image peut aider à clarifier les paramètres utilisés par le sous-fichier. Le fichier image doit être dans le même répertoire que le sous-fichier et doit avoir le même nom avec une extension appropriée au fichier image, ex: le sous-fichier exemp.ngc doit être accompagné par l'image exemp.png. Ngcgui tente de redimensionner de grandes images par sous-échantillonnage à une largeur maximale de 320 et une hauteur maximum de 240 pixels.

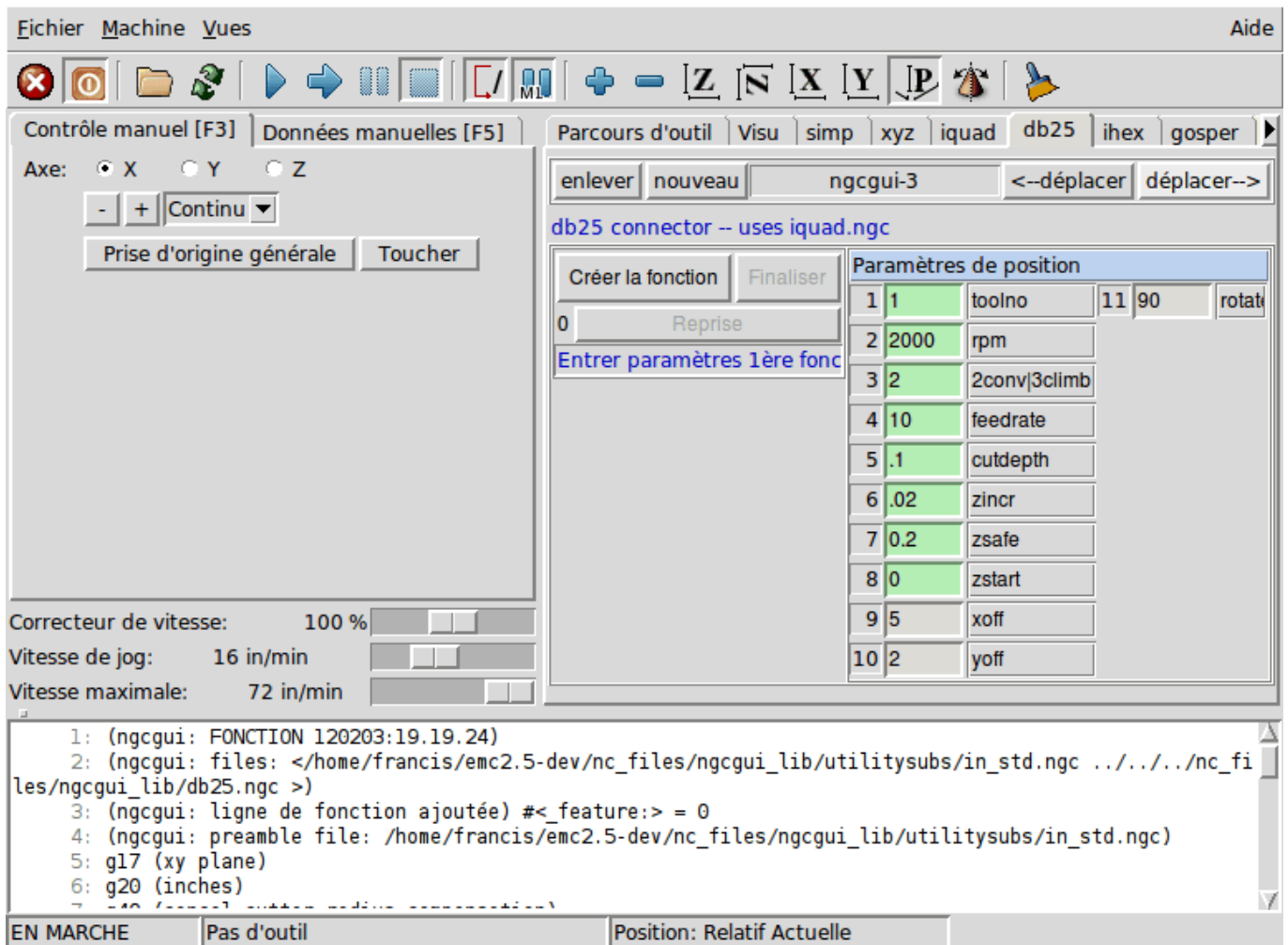
Aucune des conventions nécessaires pour faire un sous-fichier compatible ngcgui n'empêche son utilisation en tant que fichier de sous-programme pour LinuxCNC.

La distribution LinuxCNC inclut une librairie (répertoire ngcgui\_lib) qui contient plusieurs exemples de sous-fichiers et de fichiers utilitaires compatibles ngcgui pour illustrer les fonctions des sous-programmes de LinuxCNC et l'usage de ngcgui.

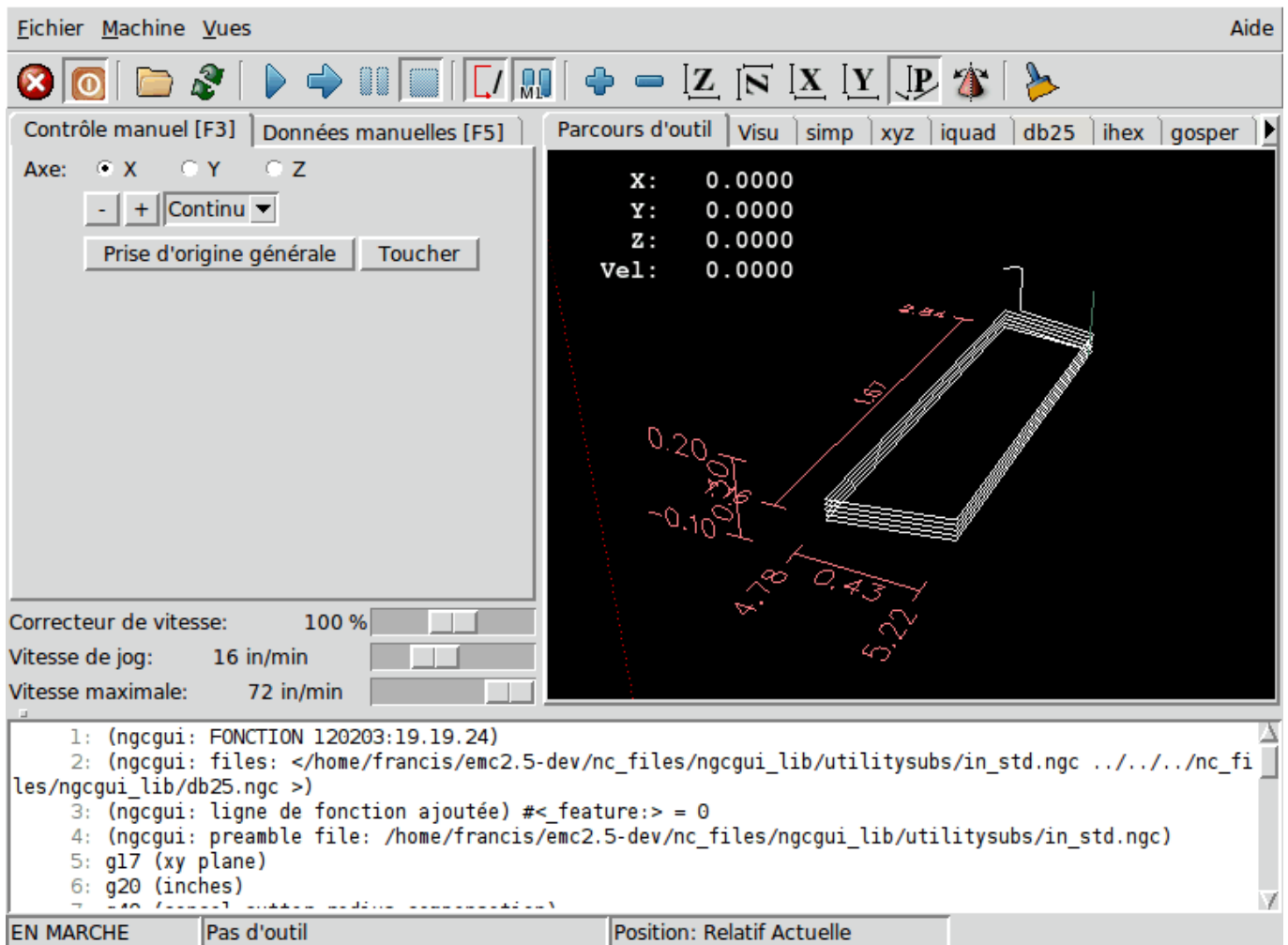
Des sous-programmes additionnels soumis par les utilisateurs se trouvent dans le forum dans la section *Subroutines*.

## 4.6 Exemple, découpe pour DB25

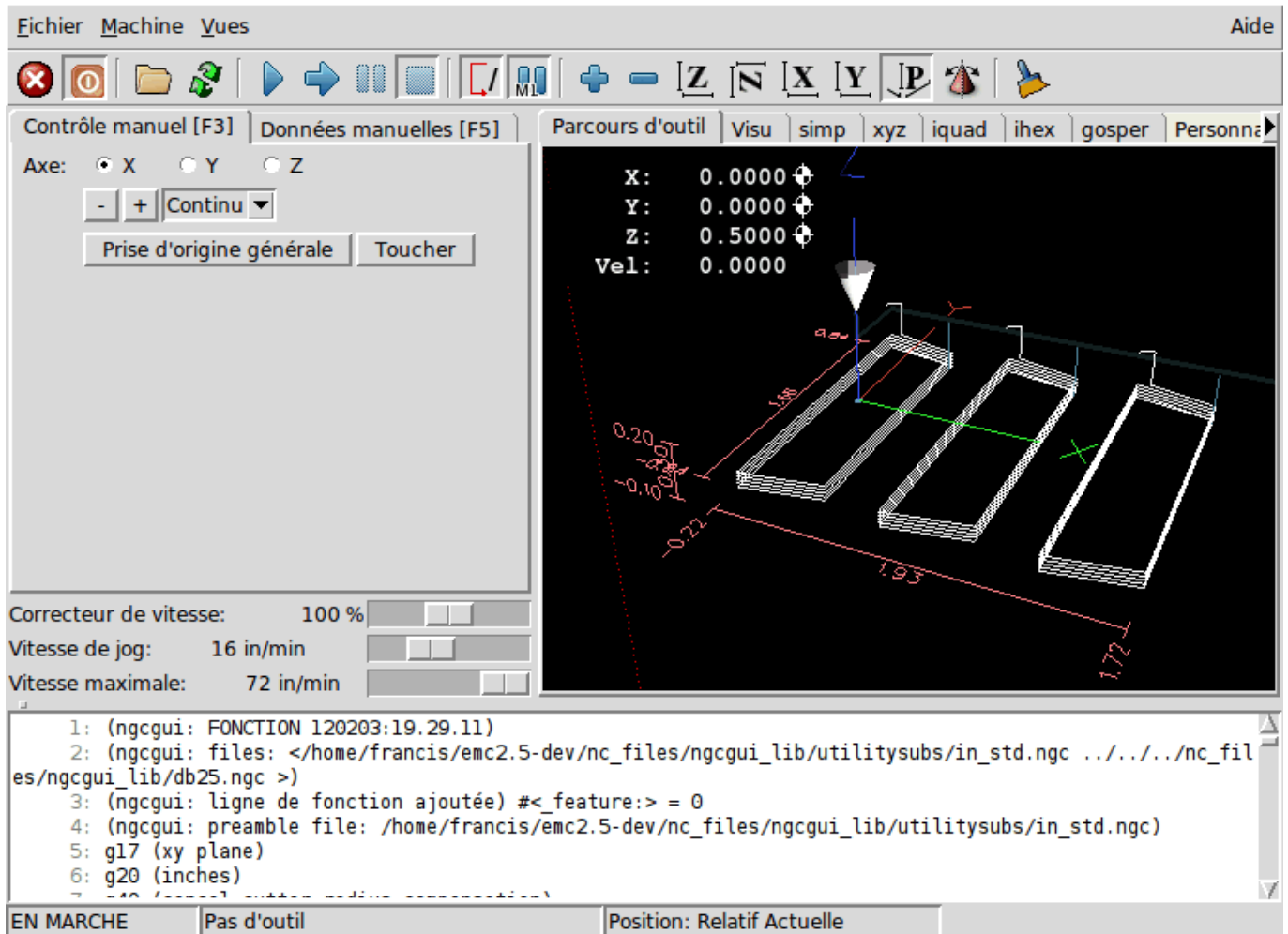
L'exemple ci-dessous montre l'utilisation du sous-programme DB25. Dans la première image on voit les champs remplis pour chaque variable.



Cette image montre le parcours d'outil du sous-programme DB25.



Cette image montre l'action du bouton *Nouveau* et de l'onglet personnalisé pour créer très facilement la découpe de trois DB25 en un seul programme.



## 4.7 Création d'un sous-programme

- Pour la création d'un sous-programme à utiliser avec Ngcgui, le nom de fichier et le nom du sous-programme doivent être les mêmes.
- Le fichier doit être placé dans le sous-répertoire pointé dans le fichier ini.
- À la première ligne peut se trouver un commentaires de type info: qui doit être placé au début du sous-programme.
- Le sous-programme doit être entouré par les balises sub et endsub.
- Les variables utilisées doivent être des variables numérotées et ne doivent pas sauter de numéro.
- Des commentaires et presets peuvent être inclus.

```

(info: simp -- simple exemple de sous-programme -- Ctrl-U pour éditer)
o<simp> sub
  #<ra>          = #1 (=0.6 Rayon A) ;Exemple de èparamtre avec un commentaire
  #<radius_b>    = #2 (=0.4)          ;Exemple de èparamtre sans commentaire
  #<feedrate>    = #3 (Feedrate)      ;Exemple de èparamtre sans preset
  g0x0y0z1
  g3 i#<ra> f#<feedrate>
  g3 i[0-#<radius_b>]
o<simp> endsub
  
```

## Chapitre 5

# L'interface graphique Touchy

Touchy est une interface utilisateur pour LinuxCNC, destinée à être utilisée avec les panneaux de commande de machines. Il ne nécessite ni clavier, ni souris.

Il a été conçu pour fonctionner également sur les écrans tactiles, en combinaison avec une manivelle électronique (MPG) et des boutons et interrupteurs.

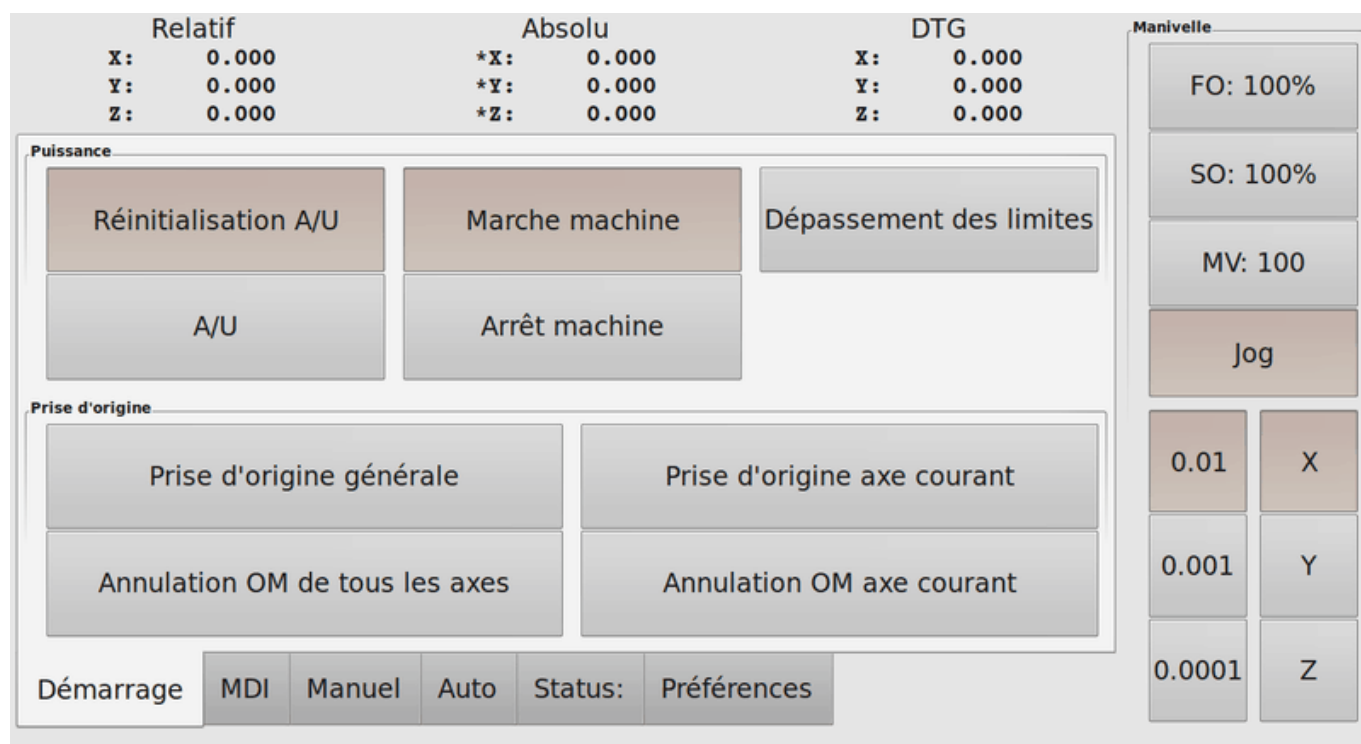


FIGURE 5.1 – L'interface tactile Touchy

## 5.1 Panneau de Configuration

### 5.1.1 Connections avec HAL

Touchy requiert qu'un fichier nommé *touchy.hal* soit créé, dans le même répertoire de configuration que le fichier ini, pour y connecter ses contrôles. Touchy exécute les commandes de HAL dans ce fichier après qu'il a créé ses propres pins, lesquelles

sont disponibles pour connexion.

Touchy dispose de plusieurs pins de sortie destinées à être connectées au contrôleur de mouvement pour gérer les manivelles de jog:

- `touchy.jog.wheel.increment`, qui doit être connecté à la pin `axis.N.jog-scale` de chacun des N axes.
- `touchy.jog.wheel.N`, qui doit être connecté à `axis.N.jog-enable` pour chacun des N axes.
- En plus d'être connecté à `touchy.wheel-counts`, le compteur d'impulsions de la manivelle doit aussi être connecté à `axis.N.jog-counts` pour chacun des N axes. Si le composant de HAL `ilowpass` est utilisé pour adoucir les mouvements de jog à la manivelle, il faut l'appliquer uniquement sur `axis.N.jog-counts` et non sur `touchy.wheel-counts`.

#### 5.1.1.1 Contrôles requis

- Un bouton *Abandon* (contact momentané) connecté sur la HAL pin `touchy.abort`
- Un bouton de *Départ cycle* (contact momentané) connecté à `touchy.cycle-start`
- Volant/Manivelle/MPG, connecté à `touchy.wheel-counts` et à la pin de mouvement comme décrit précédemment.
- Un bouton à bascule simple (contact à deux positions) connecté à `touchy.single-block`

#### 5.1.1.2 Contrôles optionnels

- Pour le jog continu, un interrupteur à trois positions avec retour au centre (ou deux boutons momentanés) pour chacun des axes concernés, attaché à `touchy.jog.continuous.x.negative` et à `touchy.jog.continuous.x.positive`, etc. pour les autres axes.
- Si un bouton de genouillère est nécessaire, (pour jogger Z en haut de sa course en grande vitesse), un bouton à contact momentané sera connecté à `touchy.quill-up`.

#### 5.1.1.3 Voyants de panneau optionnels

- `touchy.jog.active` indique quand les contrôles de jog du panneau sont actifs.
- `touchy.status-indicator` est allumé en continu quand la machine exécute un G-code et clignote quand la machine est en marche mais en pause, ou en vitesse à zéro.

### 5.1.2 Recommandé pour toutes les configurations

- Un bouton d'Arrêt d'Urgence (A/U) câblé physiquement, dans la chaîne d'arrêt d'urgence.

## 5.2 Réglages

Pour utiliser Touchy, dans la section [DISPLAY] du fichier ini de la machine, modifier la ligne de cette manière: `DISPLAY = touchy`

Quand Touchy démarre pour la première fois, vérifier l'onglet *Préférences*. Dans le cas d'un écran tactile, choisir de cacher le pointeur dans les options, pour obtenir les meilleurs résultats.

La fenêtre d'état est fixée en haut, ajustée par la taille d'une police fixe. La résolution de Gnome peut affecter cela. Si le bas de l'écran est coupé, aller dans le gestionnaire de résolution de Gnome pour revenir au réglage d'origine, à 96 DPI.

### 5.2.1 Macros

Touchy peut invoquer les macros avec mots O au travers de l'interface MDI. Pour configurer cette possibilité, dans la section [TOUCHY] du fichier ini, ajouter une ou plusieurs lignes avec `MACRO`. Chacune de ces lignes doit respecter le format suivant:

```
MACRO=increment xinc yinc
```

Dans lequel, *increment* est le nom de la macro, laquelle accepte deux paramètres, nommés ici *xinc* et *yinc*.

Maintenant, placer la macro proprement dite dans un fichier nommé *increment.ngc*. La variable `PROGRAM_PREFIX` du fichier ini pourra être utilisée pour identifier le répertoire contenant ce fichier. Il est également possible de le déclarer dans la variable `SUBROUTINE_PATH`.

Elle pourrait ressembler à cela:

```
O<increment> sub
G91 G0 X#1 Y#2
G90
O<increment> endsub
```

Noter que le nom du sous-programme, le nom de la macro ainsi que le nom du fichier .ngc doivent correspondre exactement, y compris les minuscules/ majuscules des noms.

Quand la macro est invoquée en pressant le bouton *Macro* dans l'onglet MDI de Touchy, il est possible de saisir des valeurs pour *xinc* et *yinc*, lesquelles seront passées à la macro comme étant respectivement **#1** et **#2**. Les paramètres laissées vides sont passés comme des valeur **0**.

Si il y a plusieurs macros différentes, presser le bouton *Macro* répétitivement pour les faire défiler.

Dans notre petit exemple, si -1 est entré pour *xinc* puis que le *Départ cycle* est pressé, un **G0**, sera invoqué, provoquant un déplacement en vitesse rapide, vers la gauche, de une unité machine.

Cette capacité d'utilisation des macros est très utile pour le palpage de contours ou d'orifices ainsi que pour d'autres opérations simples pré-configurées, de fraisage ou de perçage, qui pourront être réalisées depuis le panneau de Touchy sans avoir, pour cela, à écrire de programme G-code.

## Chapitre 6

# L'interface graphique TkLinuxCNC

### 6.1 Introduction

TkLinuxCNC est l'interface utilisateur graphique la plus populaire après Axis, c'est l'interface traditionnelle de LinuxCNC. Elle est écrite en Tcl et utilise le toolkit Tk pour l'affichage. Le fait d'être écrite en TCL la rend vraiment très portable (elle fonctionne sur une multitude de plateformes).



FIGURE 6.1 – L’affichage de TkLinuxCNC

## 6.2 Utiliser TkLinuxCNC

Pour sélectionner l’interface graphique TkLinuxCNC avec LinuxCNC, éditer le fichier `.ini` et dans la section `[DISPLAY]` modifier l’affichage comme ci-dessous:

```
DISPLAY = tklinuxcnc
```

Puis, lancer LinuxCNC et choisir ce fichier `ini`. La configuration qui se trouve dans `sim/tklinuxcnc/tklinuxcnc.ini` est déjà configurée pour utiliser TkLinuxCNC comme interface utilisateur.

Quand LinuxCNC est lancé avec TkLinuxCNC, une fenêtre [comme celle-ci s’affiche](#).

### 6.2.1 Une session typique avec TkLinuxCNC

1. Lancer LinuxCNC et sélectionner un fichier de configuration.
2. Libérer l’*Arrêt d’Urgence* et mettre la machine en marche (en pressant F1 puis F2).
3. Faire l’*Origine Machine* de chacun des axes.
4. Charger un fichier d’usinage.
5. Brider le brut à usiner sur la table.

6. Faire l'*Origine Pièce* de chacun des axes, à l'aide du jog ou en introduisant une valeur de décalage d'origine après un clic droit sur le nom d'un axe.
7. Lancer le programme.
8. Pour refaire une autre pièce identique, reprendre à l'étape 6. Pour usiner une pièce différente, reprendre à l'étape 4. Quand c'est terminé, quitter LinuxCNC.

## 6.3 Éléments affichés par TkLinuxCNC

La fenêtre TkLinuxCNC contient les éléments suivants:

- Une barre de menu permettant diverses actions;
- Un jeu de boutons permettant d'agir sur le mode de travail, Marche/Arrêt de la broche et autres éléments;
- Une barre de statut pour l'affichage des différents offsets;
- Une zone d'affichage des coordonnées;
- Un jeu de curseurs pour contrôler la *vitesse de jog*, le *Correcteur de vitesse d'avance* et le *Correcteur de vitesse broche* qui permettent d'augmenter ou de diminuer ces vitesses ;
- Une boîte d'entrée de données manuelles;
- Une barre de statut affichant le bloc de programme actif, G-codes, M-codes, mots F et S;
- Les boutons relatifs à l'interpréteur;
- Une zone d'affichage de texte montrant le G-code du programme chargé.

### 6.3.1 Boutons principaux

Dans la première ligne de la gauche vers la droite et cycliquement:

1. Marche Machine: *Arrêt d'Urgence Arrêt d'Urgence relâché / Marche*
2. Bascule gouttelettes
3. Broche moins vite
4. Direction de rotation de la broche *Arrêt broche / Broche sens horaire / Broche sens anti-horaire*
5. Broche plus vite
6. Annuler

puis dans la deuxième ligne:

1. Mode de marche: *MANUEL / MDI / AUTO*
2. Bascule d'arrosage
3. Bascule du contrôle frein de broche

### 6.3.2 Barre de statut des différents offsets

Elle affiche, l'offset de rayon de l'outil courant (sélectionné avec Txx M6), l'offset éventuel de longueur d'outil si il est actif et les offsets de travail (ajustables par un clic droit sur les coordonnées).

### 6.3.3 Zone d'affichage des coordonnées

La partie principale affiche la position courante de l'outil. La couleur varie selon l'état de l'axe. Si l'axe n'est pas référencé il est affiché en caractères jaunes. Si il est référencé il s'affiche en vert. Si il est en erreur, TkLinuxCNC l'affiche en rouge pour montrer un défaut. (par exemple si un contact de fin de course est activé).

Pour interpréter correctement les différentes valeurs, se référer aux boutons de droite. Si la position est *Machine*, alors la valeur affichée est en coordonnées machine. Si elle est *Relative*, la valeur affichée est en coordonnées pièce. Deux autres en dessous indiquent *actuelle* ou *commandée*. Actuelle fait référence aux valeurs retournées par les codeurs (si la machine est équipée de servomoteurs) et *commandée* fait référence à la position à atteindre envoyée aux moteurs. Ces valeurs peuvent différer pour certaines raisons: Erreur de suivi, bande morte, résolution d'encodeur ou taille de pas. Par exemple, si un mouvement est commandé

vers X0.08 sur une fraiseuse, mais qu'un pas moteur fait 0.03, alors la position *Commandée* sera 0.03 mais la position *Actuelle* sera soit 0.06 (2 pas) soit 0.09 (3 pas).

Deux autres boutons permettent de choisir entre la vue *Articulation* et la vue *Globale*. Cela a peu de sens avec les machines de type normal (cinématiques triviales), mais se révèle très utile sur les machines avec des cinématiques non triviales telles que les robots ou plateforme de Stewart. (Des informations plus complètes se trouvent dans le manuel de l'intégrateur).

### 6.3.3.1 Parcours d'outil

Quand la machine se déplace, elle laisse un tracé appelé parcours d'outil. La fenêtre d'affichage du parcours d'outil s'active via le menu *Vues* → *Parcours d'outil*.

## 6.3.4 Contrôle en automatique



FIGURE 6.2 – Interpréteur de TkLinuxCNC

### 6.3.4.1 Boutons de contrôle

Les boutons de contrôle de la partie inférieure de TkLinuxCNC, visibles sur l'image ci-dessus, sont utilisés pour l'exécution du programme:

- *Ouvrir* pour charger un fichier,
- *Lancer* pour commencer l'usinage,
- *Pause* pour stopper temporairement l'usinage,
- *Reprise* pour reprendre un programme mis en pause,
- *Pas à pas* pour avancer d'une seule ligne de programme,
- *Vérifier* pour vérifier si il contient des erreurs,
- *Arrêt optionnel* pour basculer l'arrêt optionnel, si ce bouton est vert l'exécution du programme est stoppée quand un code M1 est rencontré.

### 6.3.4.2 Zone texte d'affichage du programme

Quand un programme est lancé, la ligne courante est affichée en surbrillance blanche. L'affichage du texte défile automatiquement pour montrer la ligne courante.

## 6.3.5 Contrôle en manuel

### 6.3.5.1 Touches implicites

TkLinuxCNC permet les déplacements manuels de la machine. Cette action s'appelle le *jog*. Premièrement, sélectionner l'axe à déplacer en cliquant dessus. Puis, cliquer et maintenir les boutons + ou - selon la direction du mouvement souhaité. Les quatre premiers axes peuvent aussi être déplacés à l'aide des touches fléchées pour les axes X et Y, Pg.préc et Pg.suiv pour l'axe Z et les touches [ et ] pour l'axe A.

Si *Continu* est activé, le mouvement sera continu tant que la touche sera pressée, si une valeur d'incrément est sélectionnée, le mobile se déplacera exactement de cette valeur à chaque appui sur la touche ou à chaque clic. Les valeurs disponibles sont:

1.0000 0.1000 0.0100 0.0010 0.0001

En cliquant le bouton *Origine* ou en pressant la touche Origine, l'axe actif est référencé sur son origine machine. Selon la configuration, la valeur de l'axe peut être simplement mise à la position absolue 0.0, ou la machine peut se déplacer vers un point spécifique matérialisé par le *contact d'origine*. Voir le manuel de l'intégrateur pour plus de détails sur les prises d'origine.

En cliquant le bouton *Dépassement de limite*, la machine permet un jog temporaire pour même si l'axe a franchi une limite d'axe fixée dans le fichier .ini. Noter que si *Dépassement de limite* est activé il s'affiche en rouge.

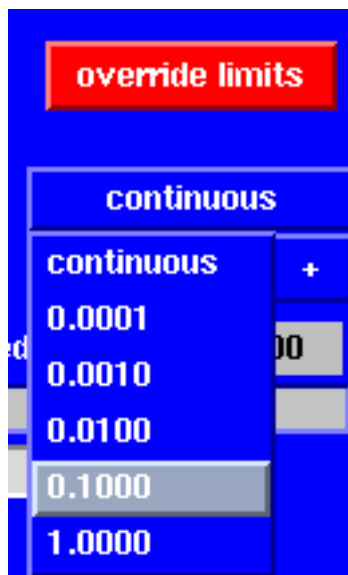


FIGURE 6.3 – Exemple de dépassement de limite et incréments de jog

### 6.3.5.2 Le groupe de boutons *Broche*

Le bouton central du dessus sélectionne le sens de rotation de la broche: Anti-horaire, Arrêt, Horaire. Les boutons fléchés augmentent ou diminuent la vitesse de rotation. Le bouton central du dessous permet d'engager ou de relâcher le frein de broche. Selon la configuration de la machine, les items de ce groupe ne sont peut être pas tous visibles.

### 6.3.5.3 Le groupe de boutons *Arrosage*

Ces deux boutons permettent d'activer ou non les lubrifiants *Gouttelettes* et *Arrosage*. Selon la configuration de la machine, les items de ce groupe ne sont peut être pas tous visibles.

## 6.3.6 Entrée manuelle de G-code (MDI)

L'entrée manuelle de données (aussi appelée MDI), permet d'entrer et d'exécuter des lignes de G-code, une à la fois. Quand la machine n'est pas en marche ni mise en mode MDI, l'entrée de code n'est pas possible.

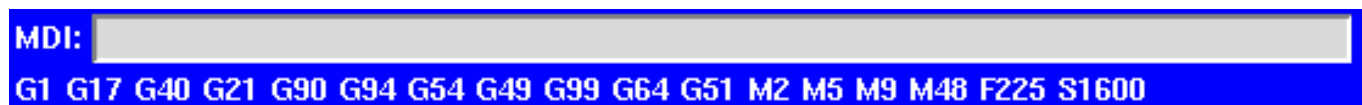


FIGURE 6.4 – Le champ de saisie des entrées manuelles

### 6.3.6.1 MDI:

Le mode MDI permet d'exécuter une commande en G-code en pressant la touche *Entrée*.

### 6.3.6.2 G-Codes actifs

Ce champs montre les *codes modaux* actuellement actifs dans l'interpréteur. Par exemple, **G54** indique que le système de coordonnées courant est celui de G54 et qu'il s'applique à toutes les coordonnées entrées.

### 6.3.7 Vitesse de Jog

En déplaçant ce curseur, la vitesse de jog peut être modifiée. Le nombre indique une vitesse en unités par minute. Le champs de texte est cliquable. Un clic ouvre un dialogue permettant d'entrer un nombre.

### 6.3.8 Correcteur de vitesse d'avance travail

En déplaçant ce curseur, la vitesse d'avance travail peut être modifiée. Par exemple, si la vitesse d'avance travail du programme est **F600** et que le curseur est placé sur 120%, alors la vitesse d'avance travail sera de 720. Le champs de texte est cliquable. Un clic ouvre un dialogue permettant d'entrer un nombre.

### 6.3.9 Correcteur de vitesse de broche

Le fonctionnement de ce curseur est le même que celui de la vitesse d'avance, mais il contrôle la vitesse de rotation de la broche. Si le programme demande S500 (broche à 500 tr/mn) et que le curseur est placé sur 80%, alors la vitesse de broche résultante sera de 400 tr/mn. Le minimum et le maximum pour ce curseur sont définis dans le fichier ini. Par défaut le curseur est placé sur 100%. Le champs de texte est cliquable. Un clic ouvre un dialogue permettant d'entrer un nombre.

## 6.4 Raccourcis clavier

La plupart des actions de TkLinuxCNC peuvent être accomplies au clavier. Beaucoup des raccourcis clavier ne sont pas accessibles en mode MDI.

Les raccourcis clavier les plus fréquemment utilisés sont montrés dans la table ci-dessous.

TABLE 6.1: Les raccourcis clavier les plus utilisés

Touche	Action
F1	Bascule de l'Arrêt d'Urgence
F2	Marche/Arrêt machine
*, 1 .. 9, 0	Correcteur vitesse d'avance 0% à 100%
X, *	Active le premier axe
Y, 1	Active le deuxième axe
Z, 2	Active le troisième axe
A, 3	Active le quatrième axe
Origine	POM de l'axe actif
Gauche, Droite	Jog du premier axe
Haut, Bas	Jog du deuxième axe
Pg.prec, Pg.suiv	Jog du troisième axe
[, ]	Jog du quatrième axe
Echap	Arrête l'exécution



## Chapitre 7

# L'interface graphique MINI

### 7.1 Introduction

1

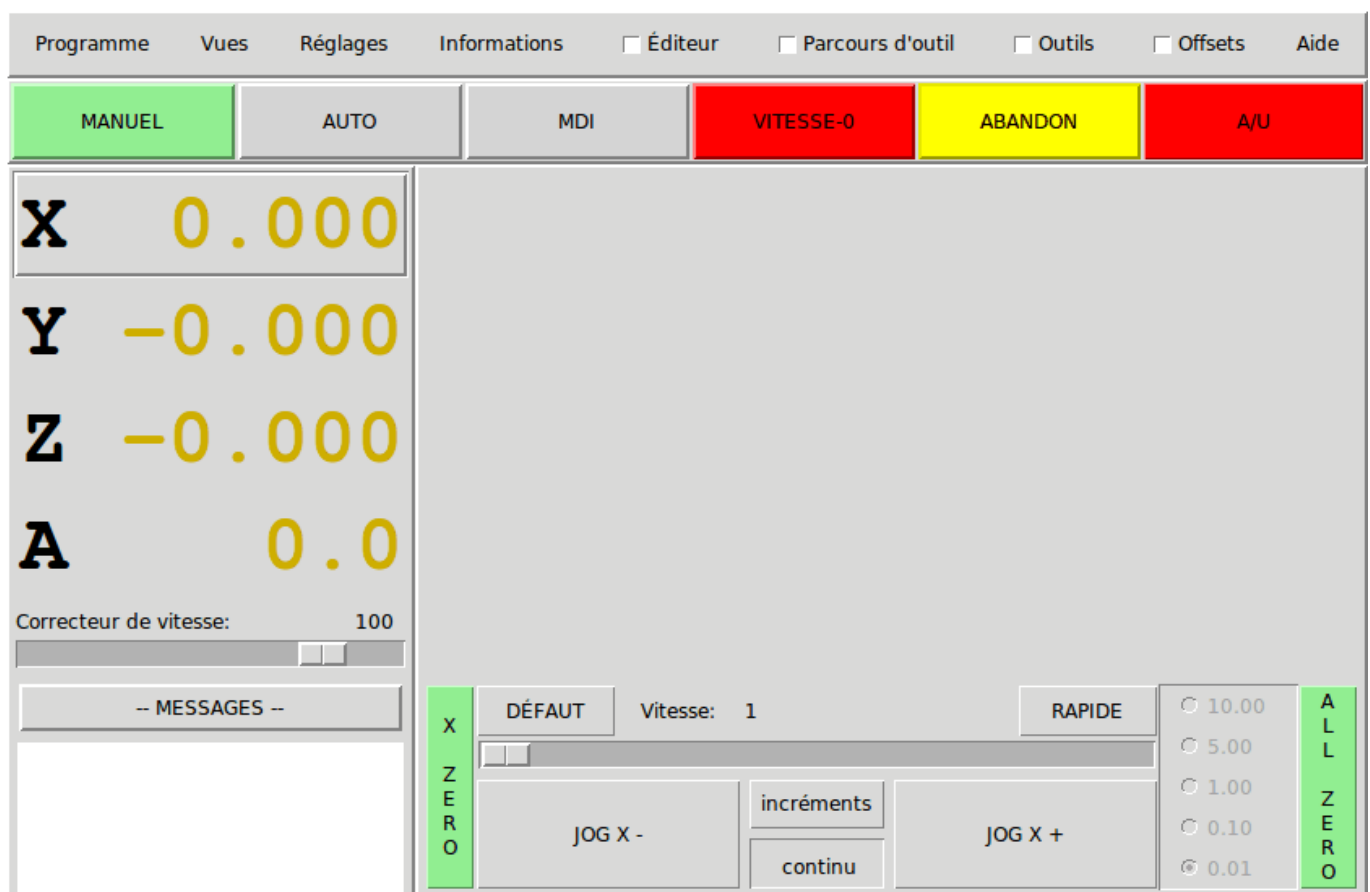


FIGURE 7.1 – L'interface graphique Mini

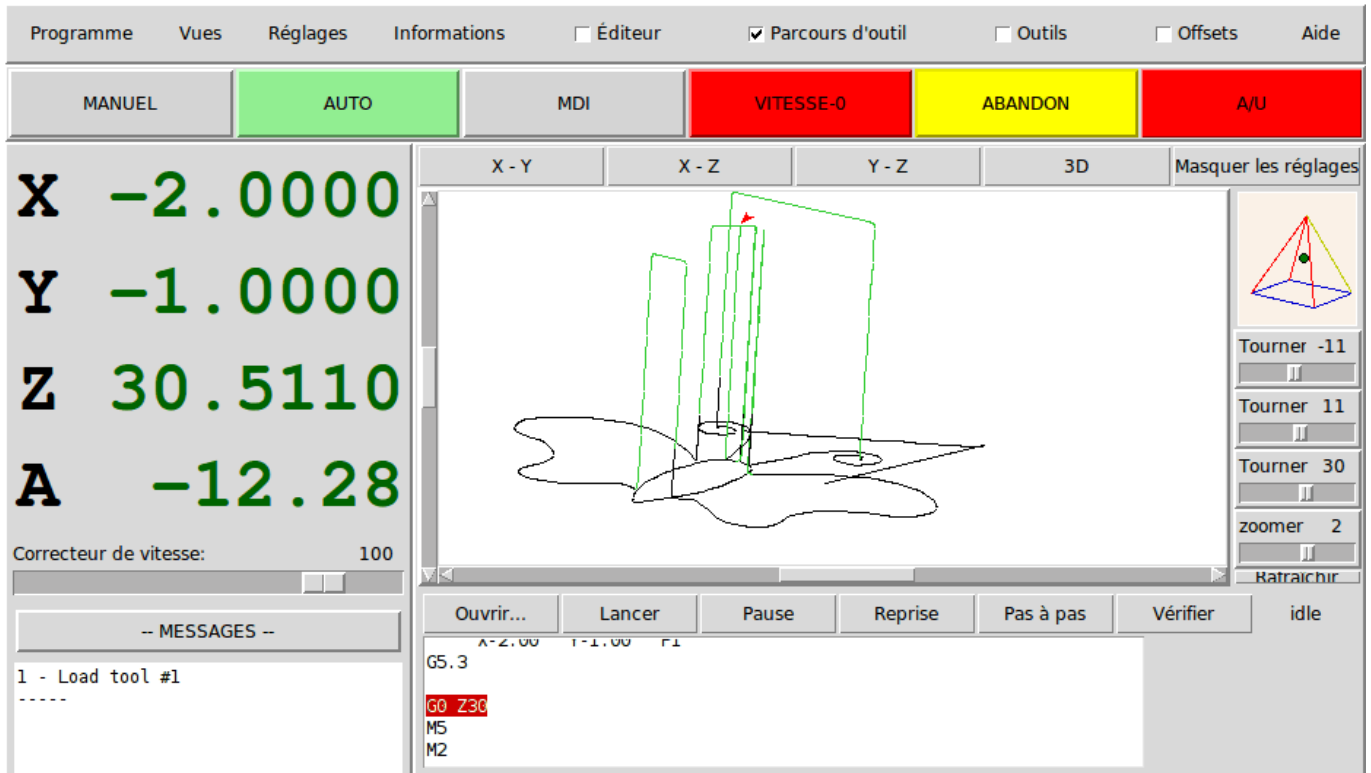
Mini a été prévu pour être une interface graphique plein écran. Il a été écrit initialement pour Sherline CNC mais est disponible pour ceux qui veulent l'utiliser, le copier le distribuer selon les termes de la GPL.

1. Une grande partie de ce chapitre provient du manuel Sherline CNC operators.

Au lieu d'ouvrir une nouvelle fenêtre pour chaque chose que l'opérateur veut faire, Mini permet de les afficher toutes dans la même fenêtre. Certaines parties de ce chapitre sont des copies des instructions écrites par Joe Martin et Ray Henry.

## 7.2 L'écran

### Affichage de Mini lors du fonctionnement de LinuxCNC



L'écran de Mini est visible dans plusieurs sections, voir la figure ci-dessus. Il inclut une barre de menu, un jeu de boutons de contrôle juste en dessous et deux larges colonnes d'information montrant l'état de la machine et permettant à l'opérateur d'entrer des commandes ou des programmes.

Quand on compare [entre l'écran d'accueil](#) et [l'écran en cours de fonctionnement](#), des différences apparaissent. Sur la seconde image:

- Chaque axe a été référencé sur son origine — L'affichage de la visu est vert sombre
- LinuxCNC est en mode auto — le bouton est vert clair
- Le parcours d'outil a été activé — le parcours est visible dans la fenêtre
- Le bloc de G-code en cours est en surbrillance blanche sur fond rouge parmi les lignes défilantes du programme.

Dès le début de l'utilisation de Mini on découvre combien il est facile de visualiser le fonctionnement de LinuxCNC et d'agir sur lui.

## 7.3 Barre de menu

La première ligne est occupée par la barre de menu. Elle permet de configurer les informations optionnelles que l'écran doit afficher. Certains des items du menu sont très différents de ce qu'il est coutume de rencontrer avec d'autres programmes. Il faut prendre quelques minutes pour regarder chaque item du menu et se familiariser avec les possibilités qu'il offre.

Le menu inclut chacune des sections et sous-sections suivantes:

### Programme

ce menu inclut les deux fonctions *reset* et *quitter*. Reset renvoie LinuxCNC dans l'état dans lequel il était quand il a démarré. Certains items de conditions de démarrage comme les unités du programme peuvent être spécifiés dans le fichier ini.

## Vues

ce menu inclus plusieurs éléments d'affichage pouvant être ajoutés pendant qu'un programme s'exécute, pour donner des informations supplémentaires. Ce qui inclus:

### **Type\_de\_position**

cet item ajoute une ligne, au dessus de l'afficheur principal de position, qui indique si la position affichée est en coordonnées machine (absolues) ou en coordonnées pièce (relatives), si la position affichée est actuelle ou commandée et si l'unité est le pouce ou le millimètre. Ce qui peut être modifié en agissant sur le menu *Réglages* décrit plus loin.

### **Informations\_outil**

cet item ajoute une ligne immédiatement sous l'afficheur principal, elle affiche quel outil est sélectionné et la valeur de l'offset appliqué.

### **Informations\_offset**

ajoute une ligne immédiatement sous celle des informations outil, elle affiche les offsets de travail actuellement appliqués. C'est la distance totale pour chaque axe depuis son point d'origine machine.

### **Afficher\_Reprise**

ajoute, en mode Auto, un bloc de boutons en bas à droite de l'affichage du programme. Ces boutons permettent à l'opérateur de relancer un programme après un A/U ou un Abandon. Le programme est figé lorsqu'un A/U ou un Abandon est engagé, mais l'opérateur peut ici choisir le bloc sur lequel redémarrer dans le mode Auto, en activant cet item de menu.

### **Masquer\_Reprise**

enlève le bloc de boutons contrôlant la reprise d'un programme stoppé par Abandon ou Arrêt d'Urgence.

### **Diviser\_la\_vue\_droite**

modifie la nature de l'affichage de la colonne de droite pour qu'il montre plus d'informations.

### **Afficher\_en\_mode\_complet**

modifie l'affichage de la colonne de droite, soit en mode boutons et la colonne divisée, soit affichage sur la colonne entière. En mode manuel et en utilisant ce mode entier, les boutons de la broche et du lubrifiant sont visibles en plus des boutons de mouvement.

### **Afficher\_dans\_toute\_la\_fenêtre**

modifie l'affichage dans la colonne de droite pour qu'elle soit remplie entièrement.

## Réglages

Les items de ce menu permettent à l'opérateur de contrôler certains paramètres en cours d'usinage.

### **Position\_Actuelle**

régle l'afficheur principal sur des valeurs de position actuelle (basées sur la machine).

### **Position\_Commandée**

régle l'afficheur principal sur des valeurs de position commandée (positions cibles).

### **Position\_Machine**

régle l'afficheur principal sur des valeurs de position absolue, distance depuis le point d'origine machine.

### **Position\_Relative**

régle l'affichage principal pour qu'il affiche les valeurs de la position courante en incluant tous les offsets comme les origines pièces qui seraient actives. Pour plus de détails voir le chapitre sur les *Systèmes de coordonnées*.

## Informations

Indique diverses activités en affichant des valeurs dans la zone MESSAGES.

### **Fichier\_programme**

Indique le nom du fichier programme actif.

### **Editeur\_de\_Fichiers**

Indique le nom de l'éditeur de texte lancé quand un fichier est choisi pour être édité.

### **Fichier\_de\_paramètres**

Indique le nom du fichier devant être utilisé pour enregistrer les paramètres du programme. Il est possible d'avoir plus d'informations dans le chapitre sur les offsets et l'utilisation des paramètres dans les programmes.

### **Fichiers\_d'outils**

Indique le nom du fichier d'outils utilisé pour l'usinage en cours.

### **G-Codes\_actifs**

Indique la liste des G-codes modaux actifs au moment où cet item a été sélectionné. Pour plus d'informations sur les codes modaux, se reporter aux chapitres sur la programmation.

## Aide

Ouvre une fenêtre de texte contenant un fichier d'aide.

Noter que entre le menu informations et le menu d'aide il y a un jeu de quatre cases à cocher. Elles sont appelées cases sélectionnables car elles passent au rouge si elles sont sélectionnées. Ces quatre cases, Editeur, Parcours d'outil, Outils et Offsets activent les différents écrans. Si plus d'une de ces cases est validée (case en rouge) il est possible de passer de l'un à l'autre des écrans surgissant par un clic droit de la souris.

## 7.4 Barre de boutons de contrôle

Sous la ligne de menu se trouve une ligne de boutons de contrôle. Ce sont les boutons principaux de l'interface. En utilisant ces boutons il est possible de changer de mode cycliquement entre *MANUEL* puis *AUTO* puis *MDI* (Manual Data Input). Ces boutons ont un fond vert clair quand le mode correspondant est actif.

Il est également possible d'utiliser les boutons *VITESSE-0*, *ABANDON* et *A/U* pour contrôler les mouvements du programme.

### 7.4.1 MANUEL

Ce bouton, ou la touche **F3** placent LinuxCNC en mode Manuel et affiche un jeu condensé de boutons que l'opérateur peut utiliser pour effectuer des commandes de mouvements manuels. Les labels des boutons de jog varient pour refléter l'axe actif. Quand en plus de ce mode manuel, la vue en mode complet est activée, les boutons de la broche et du lubrifiant sont visibles en plus de ceux de mouvement. La touche **i** ou **I** basculera entre jog en continu et jog par incréments. Presser cette touche plusieurs fois provoquera le changement cyclique de la taille de l'incrément de jog.



FIGURE 7.2 – Boutons du mode Manuel

Un bouton a été ajouté pour désigner la position actuelle comme étant la position d'origine. Une machine simple telle que la Sherline 5400 serait facile à utiliser sans avoir de position d'origine. Ce bouton met les offsets de tous les axes à zéro et place l'origine de tous les axes au point courant.

Ce concentrer sur les axes est important ici. Noter que sur la capture [du début de chapitre](#) qui est en mode manuel, on voit un cadre autour de l'axe X pour mettre son affichage en évidence. Ce cadre (le focus) indique que l'axe X est l'axe actif. Il est l'axe cible pour les mouvements de jog faits par appui sur les boutons *plus* ou *moins*. Il est possible de changer l'axe actif en cliquant sur l'affichage d'un autre axe. C'est également possible en mode manuel en pressant la touche de son nom sur le clavier. La casse n'a pas d'importance **Y** ou **y** donneront le focus à l'axe Y. **A** ou **a** le donneront à l'axe A. Pour aider à se rappeler quel est l'axe actif avant de faire un jog, son nom est indiqué sur les boutons de jog.

LinuxCNC peut faire un jog (mouvement d'un axe particulier) aussi longtemps que le bouton est maintenu pressé, si le jog est réglé sur *continu*, ou il peut se déplacer d'une valeur prédéfinie, quand il est réglé sur *incrément*s. Il est aussi possible de faire un jog de l'axe actif en pressant les touches **+** ou **-** du clavier. De nouveau, la casse n'a pas d'importance pour les jogs au clavier. Les deux petits boutons situés entre les deux gros du jog, permettent de choisir quel type de jog est souhaité. Quand on est en mode incréments, les boutons de taille d'incrément sont accessibles. La sélection d'une taille d'incrément se fait par clic sur la case à cocher avec la souris ou cycliquement en pressant la touche **i** ou **I** sur le clavier. Le jog par incréments présente quelques effets intéressants autant qu'inattendus. Si vous pressez le bouton de jog alors qu'un mouvement de jog est déjà en cours, la

distance à laquelle il était lorsqu'est arrivée la commande du second jog, sera ajoutée à la position. Deux pressions successives rapides de 10mm d'incrément ne donneront pas 20mm de mouvement. Vous devez attendre que le premier soit terminé avant d'envoyer le second.

La vitesse de jog est affichée au dessus du curseur. Il est possible de régler le curseur en cliquant dans la glissière du curseur, du côté où vous voulez son déplacement, ou en cliquant sur les boutons *Défaut* ou *Rapide*. Ces réglages n'affectent que les mouvements de jog en mode manuel. Tant qu'un mouvement de jog est en cours, un changement de la vitesse de jog est sans effet sur le jog. Par exemple, disons que vous avez réglé le jog par incréments de 10mm. Même si vous pressez le bouton *Jog*, le déplacement de dix millimètres se finira à la vitesse à laquelle il a commencé.

## 7.4.2 AUTO

Quand le bouton Auto est pressé, ou **F4** sur le clavier, LinuxCNC passe dans ce mode, un jeu de boutons traditionnels des opérations en auto est affiché et une petite fenêtre textuelle s'ouvre, montrant une partie du programme. Durant un usinage, la ligne active est affichée en surbrillance blanche sur fond rouge.

Dans le mode auto, beaucoup de touches sont liées aux contrôles. Par exemple, les touches numérotées au dessus du clavier sont liées aux réglages du correcteur de vitesse d'avance travail. Le **0** l'ajuste à 100%, le **9** l'ajuste à 90% etc. D'autres touches fonctionnent de la même manière qu'avec l'interface TkLinuxCNC.

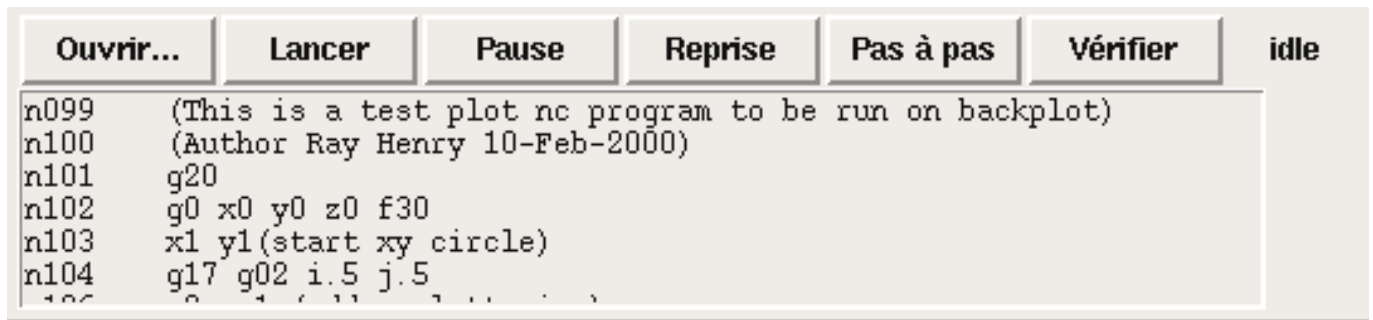


FIGURE 7.3 – Mode Auto

Le mode Auto n'affiche normalement pas les G-codes actifs ou modaux. Si l'opérateur veut les voir, il doit utiliser le menu *Informations* → *G-codes actifs* et afficher ainsi la liste des codes modaux dans la zone de texte MESSAGES.

Si un Arrêt d'Urgence ou un Abandon est pressé pendant un usinage, un jeu de boutons s'affiche en bas et à droite, ils permettent à l'opérateur de décaler la ligne de reprise vers l'avant ou vers l'arrière. Si la ligne de reprise n'est pas la dernière ligne active, elle sera mise en surbrillance blanche sur fond bleu. ATTENTION, une vitesse très faible et un doigt collé sur le bouton de pause est prudent pendant toute reprise de programme!

Ce qui est le mieux avec une machine CNC, c'est le mode auto. Le mode auto affiche les fonctions typiques que tout le monde espère utiliser avec LinuxCNC. Au dessus de la fenêtre un jeu de boutons qui contrôlent ce qui se passe en mode auto. En dessous, la fenêtre montrant la partie du programme en cours d'exécution. Quand le programme est lancé, la ligne active s'affiche en surbrillance blanche sur fond rouge. Les trois premiers boutons, *Ouvrir*, *Lancer* et *Pause* font ce que leurs noms indiquent. *Pause* stoppe le programme en cours d'exécution là où il est. Le bouton suivant, *Reprise*, reprend le mouvement. Le résultat est le même si le bouton *Vitesse-0* était pressé au lieu du bouton *Pause*, le mouvement est stoppé, *Pas à pas* reprend aussi le mouvement, mais il ne continue que jusqu'à la fin du bloc courant. Presser une nouvelle fois *Pas à pas* exécutera le mouvement du bloc suivant. Presser *Reprise* à ce moment là et l'interpréteur reviendra en arrière lire et relancer le programme. La combinaison entre *Pause* et *Pas à pas* marche un peu comme un seul bloc dans plusieurs interpréteurs. Avec la différence que *Pause* ne laisse pas le mouvement se poursuivre jusqu'à la fin du bloc courant. Le correcteur de vitesse d'avance travail peut se révéler très pratique pour s'approcher de la matière pour un premier usinage. Le placer sur 100% pour les déplacements rapides, le régler sur 10% et basculez entre *Vitesse-0* et 10% en utilisant le bouton *Pause*. Quand vous êtes satisfait et que les choses se présentent bien, pressez le zéro à la droite du neuf en haut du clavier, et c'est parti.

Le bouton *Vérifier* passe le code dans l'interpréteur sans production de mouvements. Si Vérifier trouve un problème il stoppe la lecture juste derrière le bloc posant problème et affiche un message. La plupart du temps vous serez en mesure de régler le

problème avec votre programme par la lecture du message et en vérifiant la ligne de code en surbrillance dans la fenêtre du programme. Certains messages, toutefois, ne sont pas d'un grand secours. Parfois vous devrez lire une ou deux lignes en avant de celle en surbrillance pour voir le problème. Occasionnellement le message fait référence à quelque chose très en avant de la ligne en surbrillance. Le plus souvent ça se produit si vous oubliez de terminer votre programme par un code correct comme %, M2, M30, ou M60.

### 7.4.3 MDI

Le bouton MDI ou la touche **F5** activent le mode d'entrée de données en manuel (Manual Data Input). Ce mode affiche un simple champ de saisie d'une ligne de texte et montre les codes modaux actuellement actifs dans l'interpréteur.

Le mode MDI vous permet d'entrer de simples blocs et de les faire exécuter par l'interpréteur comme si ils étaient une partie d'un programme (Une sorte de programme d'une seule ligne). Vous pouvez exécuter des cercles, des arcs, des lignes et autres. Vous pouvez aussi mettre au point une ligne de programme en entrant cette ligne comme un seul bloc, attendre que le mouvement se termine et entrer le bloc suivant. Sous la fenêtre d'entrée, se trouve une liste des codes modaux courants. Cette liste peut être très pratique. J'oublie parfois d'entrer un G0 avant une commande de mouvement. Si rien ne se passe je regarde dans la liste si G80 est actif. G80 stoppe tous les mouvements. Si il y est je me rappelle de mettre un bloc comme G00 X0 Y0 Z0. Dans le MDI, vous entrez du texte depuis le clavier, ainsi toutes les touches principales ne fonctionnent pas comme raccourcis clavier pour les commandes machine. **F1** engage l'Arrêt d'Urgence.

Puisque les touches du clavier sont nécessaires à la saisie du texte, beaucoup des raccourcis clavier disponibles en mode auto ne le sont pas ici.

### 7.4.4 VITESSE-0 et CONTINUER

Vitesse-0 est une bascule. Quand LinuxCNC est prêt pour exécuter, ou qu'il exécute une commande de mouvement, ce bouton affiche son label *VITESSE-0* sur fond rouge. Si Vitesse-0 a été pressé il affiche le label *CONTINUER*. Utiliser ce bouton pour faire une pause dans un mouvement présente l'avantage d'être capable de relancer le programme d'où il a été stoppé. Vitesse-0 bascule simplement entre vitesse zéro et la vitesse d'avance travail avec l'éventuel correcteur qui était actif au moment de l'arrêt. Ce bouton et la fonction qu'il active sont également liés à la touche pause de la plupart des claviers.

### 7.4.5 ABANDON

Le bouton Abandon stoppe tous les mouvements quand il est pressé. Il désactive aussi la commande de marche de LinuxCNC. Plus aucun mouvement ne survient après l'appui sur ce bouton. Si vous êtes en mode auto, ce bouton enlève le reste du programme du planificateur de mouvements. Il enregistre aussi le numéro de la ligne qui s'exécutait quand il a été pressé. Vous pouvez vous servir de ce numéro de ligne pour redémarrer le programme après avoir supprimé la raison de l'appui...

### 7.4.6 Arrêt d'Urgence

Le bouton d'Arrêt d'Urgence est une bascule mais il a trois fonctionnements possibles.

- Au démarrage de Mini c'est un bouton avec le texte *A/U* écrit en noir sur fond rouge. C'est état de la machine est correct pour charger un programme ou faire un jog sur un axe. L'Arrêt d'Urgence est libéré quand il s'affiche dans cet état.
- Si vous pressez sur le bouton d'Arrêt d'Urgence pendant qu'un mouvement est exécuté, le texte sur le bouton devient *A/U Engagé* sur fond gris et le bouton s'enfonce. Plus aucun mouvement n'est possible et plus rien ne réagit sur l'interface Mini tant que l'Arrêt d'Urgence est dans cet état. Le presser à nouveau à la souris le fera repasser en conditions normales de fonctionnement.
- Un troisième état est encore possible. Un bouton enfoncé portant le texte *A/U Libéré* sur fond vert signifie que l'A/U a bien été libéré mais que la machine n'a pas été mise en marche. Normalement cet état apparaît quand l'A/U était libéré mais que la touche Marche Machine **F2** a été pressée.

Joe Martin disait, *Quand tout le reste a échoué, pressez un Arrêt d'Urgence software*. Si vous avez un circuit externe qui gère l'A/U en surveillant une broche du port parallèle ou celle d'une carte d'entrées/sorties, un arrêt d'urgence software pourra couper la puissance sur les moteurs.

La plupart du temps, quand un Abandon ou un Arrêt d'Urgence est engagé c'est parce que quelque chose se passe mal. Peut être un outil cassé et qui doit être changé. On passe alors en mode manuel et on arrête la broche, on change l'outil et en supposant

que sa longueur reste la même, on est prêt pour relancer le programme. Si on renvoie l'outil à la même place ou il était avant, LinuxCNC va fonctionner parfaitement. Il est aussi possible de se déplacer sur la ligne suivante ou précédente de celle où c'est produit l'abandon. Si vous pressez le bouton *Arrière* ou *Avant* vous voyez une ligne en surbrillance bleue montrant l'écart entre la ligne sur laquelle l'abandon s'est produit (restée en surbrillance rouge) et la ligne à laquelle LinuxCNC va redémarrer. En réfléchissant à ce qui va se produire au moment de la reprise vous serez en mesure de placer l'arrêt de l'outil là où la reprise pourra se faire de manière acceptable. Vous aurez peut être à solutionner certaines difficultés comme celles créées par les compensations de rayon d'outil le long d'une ligne diagonale et vous devrez être sûr de vous avant de presser sur le bouton *Reprise*.

## 7.5 Colonne de gauche

Il y a deux colonnes sous la barre de contrôle. La colonne de gauche affiche les informations intéressant l'opérateur. Il y a seulement deux boutons dans cette zone.

### 7.5.1 Afficheurs de position des axes

Ces afficheurs se comportent exactement comme ceux de TkeLinuxCNC. La couleur des afficheurs est importante.

- Rouge, elle indique que la machine est en appui sur un contact de fin de course ou que la polarité d'une limite est mal positionnée dans le fichier ini.
- Jaune, elle indique que la machine est prête pour faire ses prises d'origine.
- Verte, elle indique que la machine a bien été référencée sur ses points d'origine.

Le type de position affichée peut varier, selon les options choisies dans le menus *Réglages*. Les réglages par défaut, ou de démarrage, peuvent être changés dans le fichier ini pour correspondre à vos besoins.

### 7.5.2 Correcteur de vitesse travail

Immédiatement sous les afficheurs de position on trouve un curseur, c'est le correcteur de vitesse travail. Vous pouvez agir sur le correcteur de vitesse et sur le bouton Vitesse-0 dans tous les modes de marche. Le correcteur agit sur la vitesse de jog et sur la vitesse d'avance travail dans les modes manuel ou MDI. Il est possible de modifier la position du curseur en le déplaçant à la souris le long de sa glissière. Il est également possible de modifier le correcteur de 1% à chaque fois qu'un clic de souris est fait dans la glissière du curseur. En mode Manuel il est possible d'ajuster le correcteur par incréments de 10% avec les touches chiffrées du haut du clavier. Le curseur est une référence visuelle très pratique pour estimer la correction appliquée sur la vitesse d'avance programmée.

### 7.5.3 Messages

L'affichage des messages situé sous le curseur du correcteur de vitesse est une sorte de bloc-notes pour LinuxCNC. Si un problème survient, il est reporté sur ce bloc-notes. Si vous essayez de déplacer un axe alors que l'Arrêt d'Urgence est engagé, vous recevez un message qui dit quelques choses à propos des conditions de marche empêchant LinuxCNC de répondre à la commande de mouvement. Si un axe est en défaut, par exemple un dépassement de limite, le message affiché sur le bloc-notes indiquera ce qui se passe. Pour demander à l'opérateur de changer d'outil par exemple, vous pouvez aussi ajouter une ligne de code dans le programme qui s'affichera sur l'écran, dans la boîte de messages. Un exemple pourrait être: (msg, Montez l'outil N°3 puis pressez Reprise). Cette ligne de code, incluse dans un programme, va afficher *Montez l'outil 3 puis pressez Reprise* dans la boîte de messages. Le mot msg, (avec la virgule) est la commande qui fait apparaître le commentaire, sans msg, le message ne serait pas affiché.

Pour effacer les messages cliquer simplement sur le bandeau --MESSAGES-- au dessus du bloc-notes ou au clavier, presser **m** tout en maintenant la touche *Alt* appuyée.

## 7.6 Colonne de droite

La colonne de droite est l'emplacement sur lequel seront affichés les différents éléments résultants des choix de l'utilisateur. C'est ici que seront visibles les différentes entrées de texte, les différents affichages et les boutons du mode manuel. C'est ici

que le parcours d'outil sera affiché pendant l'exécution d'un programme. C'est également ici que l'éditeur de texte s'ouvrira pour permettre l'édition des programmes, l'édition des tables d'outils ou d'offsets. L'écran du mode manuel a déjà été décrit précédemment. Chaque écran surgissant va être décrit en détail ci-dessous.

### 7.6.1 Editeur de texte

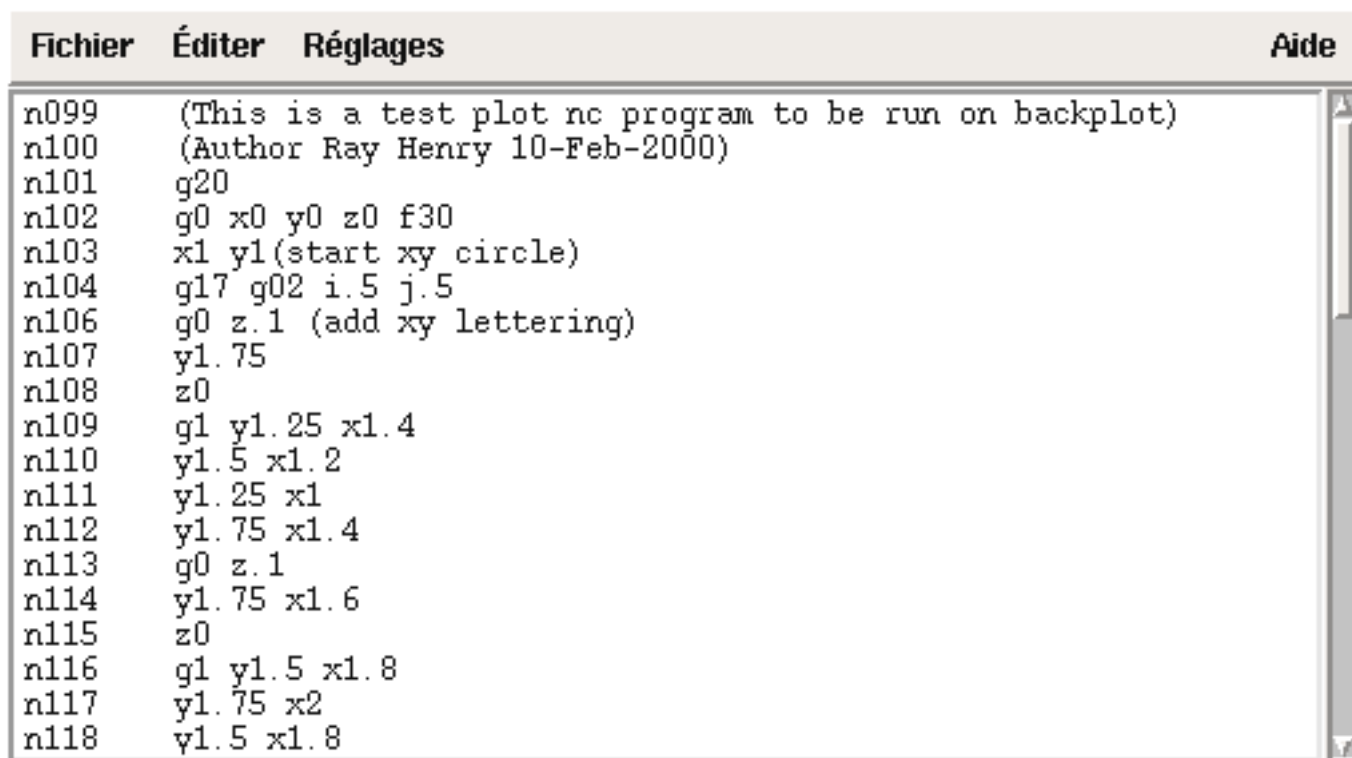


FIGURE 7.4 – Editeur de texte de Mini

L'éditeur de texte de Mini peut sembler un peu limité par rapport aux éditeurs de texte modernes. Il a été inclus parce qu'il comporte l'excellente possibilité de numéroter et renuméroter un programme de la même manière que l'interpréteur le fait avec un fichier. Il permet également le couper/coller d'une partie vers une autre du fichier. En plus, il permet d'enregistrer les changements faits dans le programme et de soumettre celui-ci à l'interpréteur de LinuxCNC depuis le même menu. Il est possible de travailler sur un fichier ouvert dans cet éditeur puis de l'enregistrer et de le recharger si LinuxCNC est en mode Auto. Si vous avez lancé un fichier et que vous avez besoin de l'éditer, ce fichier sera placé dans l'éditeur quand vous cliquez sur le bouton *Editeur* du menu.

## 7.6.2 Parcours d'outil

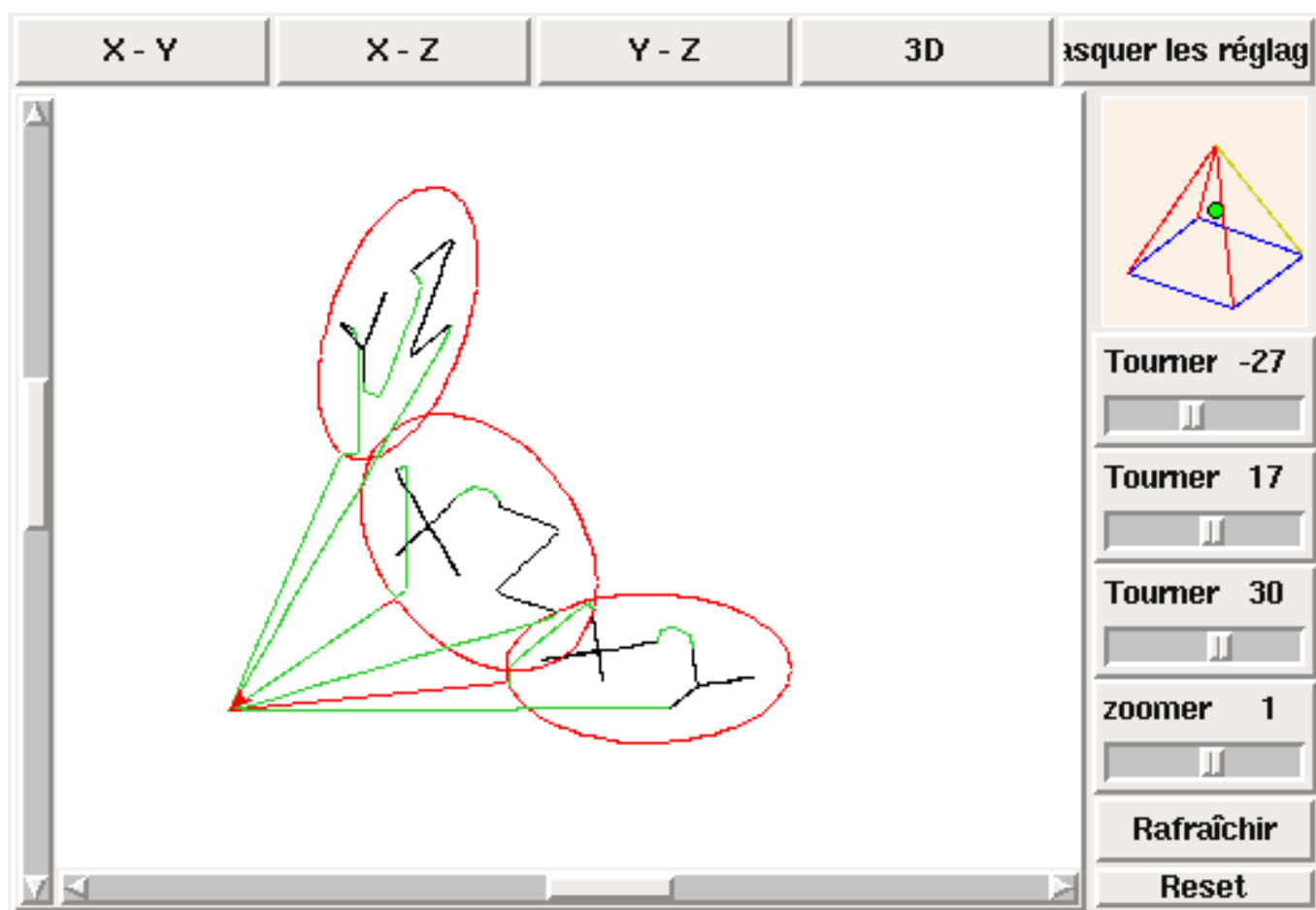


FIGURE 7.5 – Parcours d'outil de Mini

Le bouton *Parcours d'outil* du menu, affiche un tracé produit par le cheminement de l'outil pendant l'usinage. Cette trace est affichable selon plusieurs plans et en **3-D** qui est le mode par défaut. Les choix sont affichés au dessus et à droite de cette fenêtre. Si l'usinage est déjà en cours depuis un certain temps quand cette fenêtre est activée, il lui faudra un peu de temps pour recalculer la vue du parcours.

Sur le côté droit de la fenêtre une petite pyramide graphique montre l'angle depuis lequel l'opérateur voit le tracé du parcours d'outil. En dessous, une série de curseurs permettent de modifier les angles et la taille de la vue. Il est possible de contrôler les modifications en regardant l'attitude de la petite pyramide. Les modifications ne prendront effet qu'après appui sur le bouton *Rafraîchir*. Le bouton *Reset* efface toutes les traces du parcours affiché et prépare la fenêtre pour un nouveau lancement du programme, il conserve toutefois les orientations et le niveau de zoom actuellement définis.

Si un parcours d'outil est lancé avant qu'un programme ne soit démarré, il va afficher quelques lignes de couleur pour indiquer le type de tracé qui sera utilisé. Une ligne verte représente un mouvement en vitesse d'avance rapide. Une ligne noire, un mouvement en vitesse d'avance travail. Une ligne bleue et une ligne rouge indiquent respectivement un arc en sens anti-horaire et un arc en sens horaire.

Le tracé du parcours d'outil de Mini permet de zoomer et d'orienter les vues du programme en cours d'exécution mais il n'est pas conçu pour enregistrer un chemin d'outil sur une longue période de temps.

### 7.6.3 Page des outils

La page des outils permet d'ajuster la longueur et le diamètre des outils, ces valeurs deviennent effectives dès l'appui sur la touche *Entrée*. Il est nécessaire de définir les paramètres des outils avant de lancer le programme. Les offsets d'outils ne peuvent pas être modifiés une fois le programme démarré ni quand il est en pause.

RÉGLAGE OUTIL			
Cliquer ou « tab » pour éditer. Presser « Entrée » pour retourner au contrôle de la machine depuis le clavier.			
NUMÉRO D'OUTIL	LONGUEUR	DIAMÈTRE	COMMENTAIRE
2	0.0	1.0	forêt de 1
3	0.0	2.0	forêt de 2
4	0.0	4.0	fraise 2 dents
Ajouter un outil supplémentaire		Enlever le dernier outil	

FIGURE 7.6 – Affichage de la table d'outils de Mini

Les boutons *Ajouter un outil* et *Enlever le dernier outil* portent sur la fin de la liste des outils de sorte que l'ajout d'outils s'effectuent vers le bas. Quand un nouvel outil est ajouté, il est utilisable dans le programme avec la commande G-code habituelle. Le nombre maximum d'outils est de 32 dans le fichier de configuration de LinuxCNC mais vous serez arrivé au bout des possibilités d'affichage de Mini bien avant cela. A la place vous pouvez utiliser le *menu* → *Vues* → *Afficher dans toute la fenêtre* pour voir plus d'outils si nécessaire.

### 7.6.4 Page des offsets

La page des offsets peut être utilisée pour afficher et ajuster les décalages d'origine des différents systèmes de coordonnées. Le système de coordonnées est choisi dans la colonne de gauche, par sélection d'une case à cocher. Quand un système de coordonnées est choisi il est possible d'entrer directement les valeurs ou de déplacer l'axe à une position d'apprentissage.

**RÉGLAGE COORDONNÉES SYSTÈME**

**Cliquer la valeur à éditer au clavier. Presser « entrée » pour revenir au contrôle de la machine depuis le clavier.**

◇ G54	Axe	Valeur	
◆ G55	X	0.000000	Apprentissage
◇ G56	Y	0.000000	Apprentissage
◇ G57	Z	0.000000	Apprentissage
◇ G58			
◇ G59	Offset par rayon	0.000000	
◇ G59.1	Offset par longueur	0.000000	
◇ G59.2	◇ Soustraire	◆ Ajouter	
◇ G59.3	mettre tous les G55 à zéro		Écrire et charger fichier

FIGURE 7.7 – Mini Offset Display

Il est également possible d'utiliser une pinnule puis d'ajouter le rayon et la longueur par le bouton *Apprentissage*. Pour cela, il peut être nécessaire d'ajouter ou de soustraire le rayon de la pinnule selon la surface qui est touchée. C'est la case à cocher *Soustraire* ou *Ajouter* qui permet ce choix.

Le bouton *Tous les Gxx à zéro* enlève tous les décalages du système courant visibles sur l'écran mais ne modifie pas les variables dans le fichier tant que le bouton *Écrire et recharger* n'est pas pressé. Le bouton *Écrire et recharger* est à presser pour valider et recharger tous les changements effectués sur les valeurs des systèmes de coordonnées.

## 7.7 Raccourcis clavier

Un certain nombre de raccourcis clavier utilisés avec TkLinuxCNC ont été conservés avec Mini. Quelques uns ont changé pour étendre leurs fonctions ou pour faciliter les opérations sur une machine utilisant cette interface. Certaines touches opèrent de la même manière quelque soit le mode. D'autres change en fonction du mode de travail de LinuxCNC.

### 7.7.1 Touches courantes

#### Pause

Bascule du correcteur de vitesse travail

#### Echap

Abandonne le mouvement

#### F1

Bascule l'état de l'Arrêt d'Urgence

#### F2

Marche/Arrêt machine

#### F3

Mode manuel

#### F4

Mode Auto

#### F5

Mode MDI

**F6**

Réinitialise l'interpréteur

Ces touches fonctionnent seulement pour une machine utilisant des entrées sorties auxiliaires

**F7**

Marche/Arrêt du brouillard

**F8**

Marche/Arrêt arrosage

**F9**

Marche/Arrêt broche sens horaire

**F10**

Marche/Arrêt broche sens anti-horaire

**F11**

Diminue la vitesse de broche

**F12**

Augmente la vitesse de broche

## 7.7.2 Mode Manuel

**1-9 0**

Positionne le correcteur de vitesse d'avance travail de 10% en 10%, de 0 à 100%

~

Positionne le correcteur de vitesse à 0

**x**

Sélectionne l'axe X

**y**

Sélectionne l'axe Y

**z**

Sélectionne l'axe Z

**a**

Sélectionne l'axe A

**b**

Sélectionne l'axe B

**c**

Sélectionne l'axe C

**Gauche Droite**

jog de l'axe X

**Haut Bas**

jog de l'axe Y

**Pg.préc Pg.suiv**

jog de l'axe Z

– \_:: jog de l'axe actif dans la direction moins

**+ =**

jog l'axe actif dans la direction plus

**Origine**

POM de l'axe actif

**i I**

Bascule cyclique des incréments de jog

Les touches suivantes ne fonctionnent que sur les machines utilisant des entrées sorties auxiliaires

**b**

Relâcher le frein de broche

**Alt-b**

Engager le frein de broche

### 7.7.3 Mode Auto

#### **1-9,0**

Positionne le correcteur de vitesse d'avance travail de 10% en 10%, de 0 à 100%

~

Positionne le correcteur de vitesse à 0

#### **o/O**

Ouvre un programme

#### **r/R**

Lance le programme ouvert

#### **p/P**

Met le programme en pause

#### **s/S**

Reprise d'un programme en pause

#### **a/A**

Avance d'une ligne dans un programme en pause

## 7.8 Divers

Une des possibilités de Mini est d'afficher n'importe quel axe, au delà de 2, comme étant un axe rotatif, il affiche alors des degrés pour celui-ci. Il converti également les unités en degrés quand un axe rotatif a le focus.

## Chapitre 8

# L'interface textuelle KEYSTICK

### 8.1 Introduction

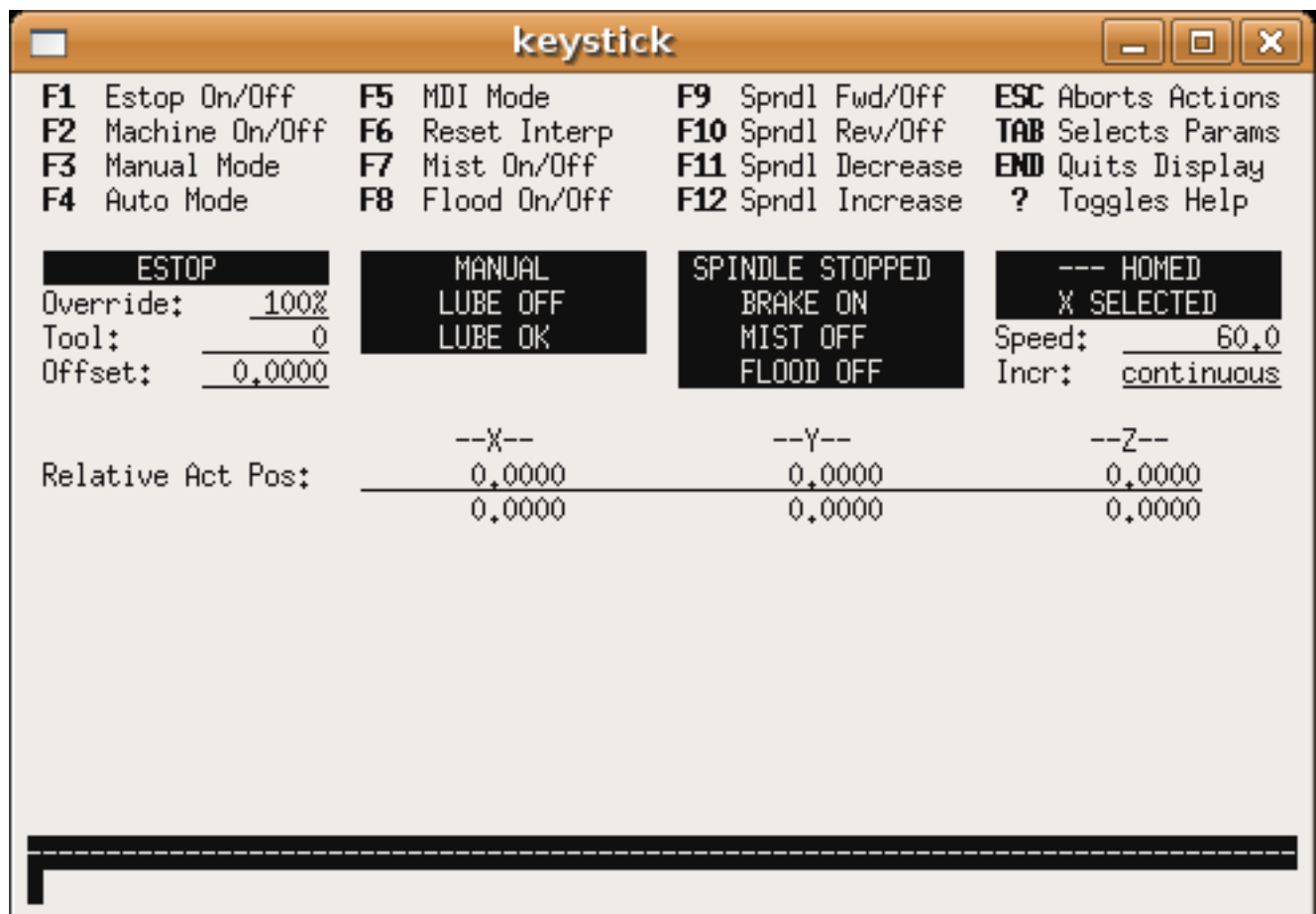


FIGURE 8.1 – L'interface utilisateur minimale

Keystick est une interface minimaliste textuelle.

## 8.2 Installation

Pour utiliser «keystick» modifier la section DISPLAY du fichier ini de la manière suivante:

```
[DISPLAY]
```

```
DISPLAY = keystick
```

## 8.3 Utilisation

Keystick est extrêmement simple à utiliser. Dans le mode MDI vous pouvez directement taper le G-code et il sera visible dans la partie supérieure. La touche «?» bascule avec ou sans aide.

1

---

1. ndt: keystick a été conçu pour être minimaliste, par conséquent son interface n'est pas traductible.

---

## **Deuxième partie**

# **L'utilisation de LinuxCNC**

---

## Chapitre 9

# Concepts importants pour l'utilisateur

### 9.1 La configuration machine

Le dessin suivant montre les directions de déplacement de l'outil et la position des fins de course de limite sur une fraiseuse classique. Noter le diagramme cartésien représentant les directions de déplacement de l'outil (Tool Direction). La direction de déplacement de la table et en opposition du système de coordonnées cartésiennes. Le système de coordonnées cartésiennes représente le sens de déplacement de l'outil. C'est toujours les déplacements de l'outil qui doivent être programmés pour que l'outil se déplace dans les directions correctes par rapport au matériel.

Noter également la position des fins de course et le sens d'activation de leurs cames. Plusieurs combinaisons sont possibles, par exemple il est possible, à l'inverse du dessin, de placer un seul fin de course fixe au milieu de la table et deux cames mobiles pour l'actionner. Dans ce cas les limites seront inversées, +X sera à droite de la table et -X à gauche. Cette inversion ne change rien du point de vue du sens de déplacement de l'outil.

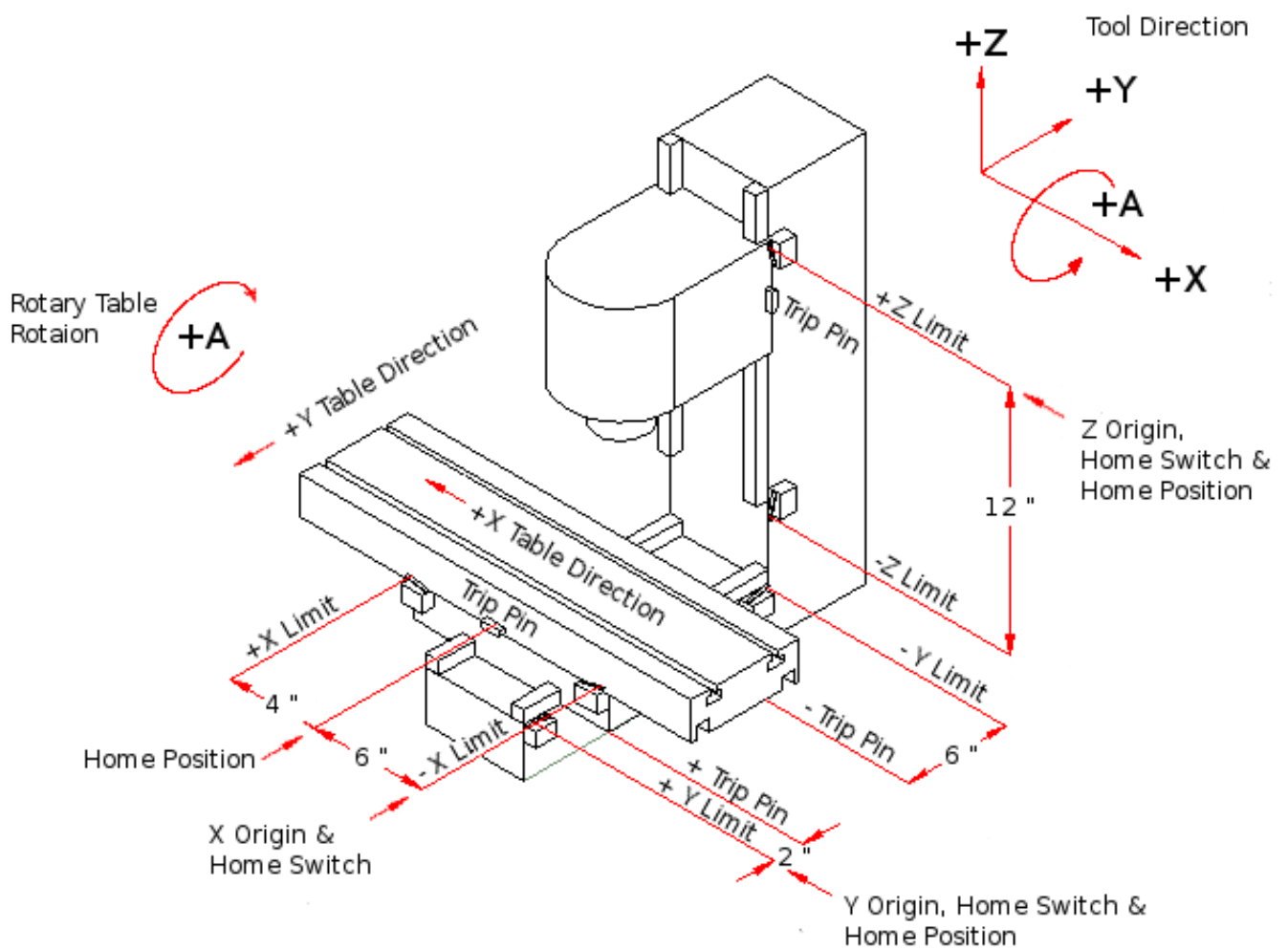


FIGURE 9.1 – Configuration typique d'une fraiseuse

Le dessin suivant montre les directions de déplacement de l'outil et la position des fins de course de limite sur un tour classique.

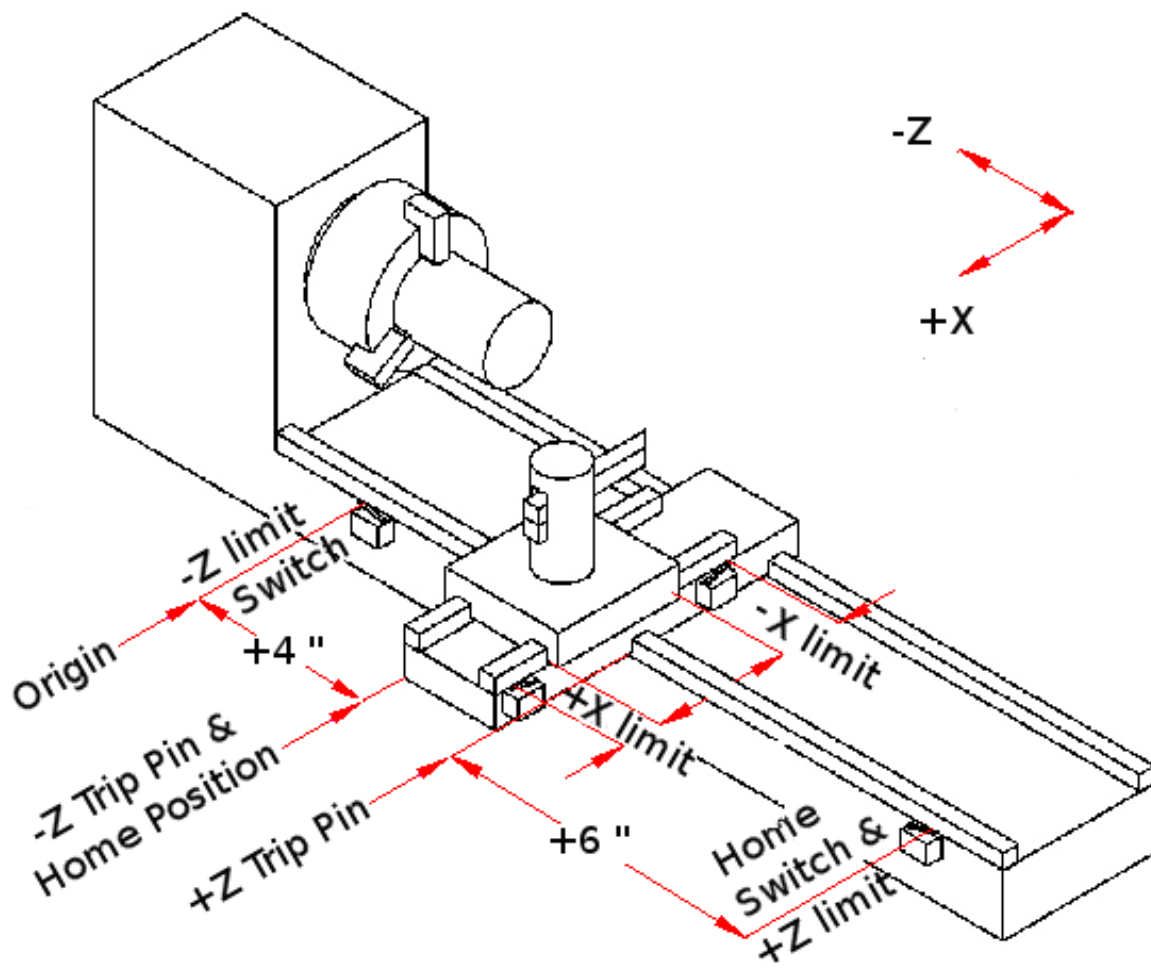


FIGURE 9.2 – Configuration typique d'un tour

## 9.2 Contrôle de trajectoire

### 9.2.1 La planification de trajectoire

La planification de trajectoire est en général, le moyen qui permet à LinuxCNC de suivre le chemin spécifié par le programme G-code, tout en restant dans les limites permises par la machine.

Un programme en G-code ne peut jamais être exactement suivi. Par exemple imaginez que vous spécifiez dans une ligne du programme les mouvements suivants:

```
G1 X10 F100 (G1 un mouvement élinaire, X10 la destination, F100 la vitesse)
```

En réalité, la totalité du mouvement ne peut pas être effectuée à F100, puisque la machine commence le mouvement à une vitesse nulle, elle doit accélérer pour se déplacer vers X=10, puis décélérer pour revenir à une vitesse nulle en fin de mouvement. Parfois une portion du mouvement se fera bien à F100, mais pour beaucoup de mouvements, spécialement les petits mouvements, la vitesse spécifiée ne sera jamais atteinte.

Les accélérations et décélérations de base décrite ici ne sont pas complexes et ne nécessite pas de compromis. Les contraintes des axes de la machine sont placés dans le fichier INI, comme la vitesse maximum de l'axe et l'accélération ne devant pas être dépassées par le planificateur de trajectoire.

## 9.2.2 Le suivi du parcours

Un problème plus compliqué est posé par le suivi du parcours. Quand vous programmez un angle droit en G-code, le planificateur de trajectoire peut suivre différents parcours, tous sont bons dans certains cas; il peut décélérer et s'arrêter exactement sur les coordonnées du sommet de l'angle, puis accélérer dans la direction perpendiculaire. Il peut également faire ce qui est appelé le mode *trajectoire continue*, qui consiste à maintenir la vitesse d'avance en passant vers le sommet de l'angle, ce qui nécessite d'arrondir l'angle de façon à respecter les contraintes machine. Vous pouvez remarquer qu'il y a dans ce cas un compromis: vous pouvez ralentir pour avoir un meilleur suivi du parcours, ou conserver une vitesse d'avance élevée au détriment de la finesse des angles, du fait d'un moins bon suivi du parcours. Selon les particularités de l'usinage, du matériau, de l'outillage, etc., le programmeur devra décider du bon compromis.

## 9.2.3 La programmation du planificateur

Les commandes de contrôle de trajectoire sont les suivantes:

### G61

(mode trajectoire exacte) G61 indique au planificateur de suivre exactement la trajectoire prévue.

### G61.1

(mode Arrêt exact) G61.1 demande au planificateur de s'arrêter exactement à la fin de chaque segment. Le parcours sera suivi avec exactitude mais les arrêts complets de l'avance peuvent se révéler destructeurs pour la pièce ou l'outillage, selon les particularités de l'usinage.

### G64

(mode trajectoire continue sans tolérance) Le mode G64 est le mode par défaut au démarrage de LinuxCNC. G64 est juste une trajectoire continue, le *Détecteur naïve CAM* n'est pas activé. G64 et G64 P0 indiquent au planificateur de sacrifier la précision de suivi du parcours pour conserver une vitesse d'avance élevée. Ce mode est nécessaire pour certains types de matériaux ou d'outillages pour lesquels l'arrêt exact est dangereux. Il peut très bien fonctionner tant que le programmeur garde à l'esprit que le parcours d'outil pourra être plus arrondi que celui indiqué par le programme. Dans le cas d'un mouvement en G0 (rapide) avec G64, faire preuve de prudence sur les mouvements de dégagement et prévoir suffisamment de distance pour éviter les obstacles selon les capacités d'accélération de la machine.

### G64 Px.xxx

(mode trajectoire continue avec tolérance) Ce mode active le *Détecteur naïve CAM* et active le mode trajectoire continue avec tolérance. Si vous utilisez le millimètre comme unité et programmez G64 P1.27, vous dites au planificateur que vous souhaitez une vitesse d'avance continue, mais qu'aux coins programmés vous voulez un ralentissement suffisant pour que le parcours de l'outil puisse rester à moins de 1.27mm du parcours programmé. L'amplitude exacte du ralentissement dépend de la géométrie de l'angle programmé et des contraintes machine, mais la seule chose dont le programmeur ait à se soucier est la tolérance, ce qui lui donne le contrôle complet des compromis du suivi de parcours. La tolérance de ce mode peut être modifiée tout au long du programme si nécessaire. Attention: spécifier un G64 P0 aura le même effet qu'un G64 seul (voir ci-dessus), c'est rendu nécessaire pour conserver la compatibilité ascendante avec les anciens programmes G-code. Voir le chapitre sur le G-code pour plus d'information sur G64 P- Q-.

### Trajectoire continue sans tolérance

Le point contrôlé touchera chaque mouvement spécifié à au moins un point. La machine ne pourra jamais se déplacer à une vitesse d'avance telle qu'elle ne puisse pas s'arrêter avec précision à la fin du mouvement en cours (ou du prochain mouvement, si vous mettez en pause lorsque la trajectoire est déjà commencée). La distance avec le point final du mouvement est aussi grande que nécessaire pour maintenir la meilleure vitesse d'avance possible pendant le parcours.

### Détecteur Naïve Cam

Les mouvements successifs en G1, concernant uniquement les axes XYZ, dont la déviation par rapport à une ligne droite est inférieur à P, sont fusionnés en une seule ligne droite. Ce mouvement fusionné remplace les mouvements individuels en G1 pour obtenir une nouvelle trajectoire avec tolérance. Entre les mouvements successifs, le point contrôlé ne passera jamais à plus de P- du point final du mouvement en cours. Le point contrôlé touchera au moins un point de chacun des mouvements. La machine ne pourra jamais se déplacer à une vitesse ne lui permettant pas de venir s'arrêter exactement à la fin du mouvement actuel (ou du prochain mouvement, si vous mettez en pause lorsque la trajectoire est déjà commencée). En mouvement G2/3 dans le plan G17 (XY) quand la déviation maximale entre un arc et une ligne droite est plus petite que la tolérance G64 Q- l'arc est brisé en deux lignes (du début de l'arc à son milieu et du milieu à la fin de l'arc). Ces deux tronçons sont ensuite soumis à l'algorithme Naïve cam des lignes. Ainsi, les cas ligne-arc, arc-arc et arc-ligne, comme les cas ligne-ligne bénéficient du traitement *Détecteur naïve CAM*. Les performances de contournage sont accrues grâce à la simplification de la trajectoire.

Dans la figure suivante la ligne bleue représente la vitesse machine actuelle. La ligne rouge représente la capacité d'accélération de la machine. La ligne horizontale sous chaque tracé est le mouvement planifié. Le tracé supérieur montre comment le planificateur de trajectoire ralenti la machine quand des petits mouvements sont rencontrés. Ceci pour rester dans les limites fixées par les paramètres d'accélération de la machine et être capable de s'arrêter exactement à la fin du prochain mouvement. Le tracé du bas montre l'effet du détecteur Naive Cam pour combiner les mouvements et fournir une amélioration conséquente dans le suivi de la vitesse programmée.

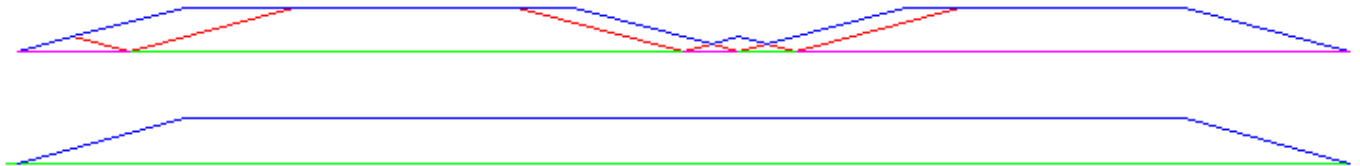


FIGURE 9.3 – Détecteur Naive Cam

## 9.2.4 Planification des mouvements

Assurez-vous que les mouvements soient *assez longs* pour convenir à votre machine/matériel. Principalement en raison de la règle selon laquelle "la machine ne pourra jamais se déplacer à une vitesse ne lui permettant pas de venir s'arrêter complètement à la fin du mouvement actuel", il y a une longueur minimale de déplacement permettant à la machine d'atteindre la vitesse demandée avec un réglage d'accélération donné.

Les phases d'accélération et de décélération utilisent chacune la moitié de la variable MAX\_ACCELERATION du fichier .ini. Avec une trajectoire continue c'est exactement inversé, ce qui fait que l'accélération totale de l'axe est égal à la variable MAX\_ACCELERATION. Dans d'autres cas, l'accélération actuelle de la machine est un peu inférieure à celle du fichier ini.

Pour maintenir la vitesse d'avance, le mouvement doit être plus long que la distance qui lui est nécessaire pour accélérer de zéro à la vitesse souhaitée, puis de décélérer pour s'arrêter. En utilisant  $A$  comme étant  $1/2$  de la variable MAX\_ACCELERATION du fichier ini et  $F$  comme étant la vitesse d'avance *en unités par seconde*, le temps d'accélération sera  $t_a = F/A$  et la distance d'accélération sera  $d_a = F * t_a / 2$ . Les temps et distance de décélération sont les mêmes, ce qui fait que la distance critique  $d = d_a + d_d = 2 * d_a = F^2/A$ .

Par exemple, pour une vitesse d'avance de  $25\text{mm par seconde}$  et une accélération de  $250\text{ mm/sec}^2$ , la distance critique sera de  $10^2/100 = 100/100 = 1\text{mm}$ . Pour une vitesse d'avance de  $5\text{mm par seconde}$ , la distance critique ne serait que de  $5^2/100 = 25/100 = 0.25\text{mm}$ .

## 9.3 G-code

### 9.3.1 Par défaut

Quand LinuxCNC démarre pour la première fois beaucoup de G et M codes sont chargés par défaut. Les codes actifs courants sont visibles dans l'interface Axis, dans l'onglet *Données manuelles* dans le champ *G-codes actifs*. Ces codes G et M définissent le comportement de LinuxCNC et il est important de bien comprendre la signification de chacun avant de démarrer LinuxCNC. Ces codes par défaut peuvent être modifiés lors du lancement d'un fichier de G-codes puis laissés dans différents états qui seront identiques lors d'une nouvelle session de LinuxCNC. La bonne pratique consiste à mettre dans le préambule de chaque fichier de G-codes les codes nécessaires pour le travail demandé et ne pas supposer que ceux par défaut conviendront. Imprimer la page des références rapides du G-code peut aider à se rappeler la signification de chacun d'eux.

## 9.4 Vitesse d'avance

Si vous avez un tour ou un axe rotatif, pour savoir comment la vitesse d'avance s'applique selon que l'axe est linéaire ou rotatif, lire et comprendre la section [vitesse d'avance](#) du manuel de l'utilisateur.

## 9.5 Compensation de rayon d'outil

La compensation de rayon d'outil (G41/G42) nécessite que l'outil puisse usiner tout au long de la trajectoire programmée sans interférer avec les mouvements d'entrée ou de sortie. Si c'est impossible avec le diamètre de l'outil courant, une erreur est signalée. Un diamètre d'outil inférieur est peut être utilisable sans erreur pour le même parcours. Ce qui signifie que quand ce type de problème se présente, il est possible de programmer un outil plus petit pour usiner le même parcours sans erreur. Voir la section compensation de rayon d'outil pour plus d'informations.

## 9.6 Prise d'origine machine

Après le démarrage de LinuxCNC chaque axe doit être référencé sur son point d'origine machine avant tout mouvement ou commande MDI.

Pour déroger à ce comportement par défaut, ou pour utiliser l'interface Mini, il est possible d'ajuster l'option NO\_FORCE\_HOMING = 1 dans la section [TRAJ] du fichier ini.

## 9.7 Changement d'outil

Il existe plusieurs options pour effectuer un changement d'outil. Voir la section [EMCIO] dans le manuel de l'intégrateur pour les informations sur la configuration de ces options. Voir également les sections G28 et G30 du manuel de l'utilisateur.

## 9.8 Systèmes de coordonnées

Les systèmes de coordonnées peuvent être déroutant au premier abord. Avant de démarrer une machine CNC, il est important de bien comprendre les bases des systèmes utilisés par LinuxCNC. Pour explorer plus en profondeur les systèmes de coordonnées utilisés par LinuxCNC, voir la section xxxxx de ce manuel.

### 9.8.1 G53 Coordonnées machine

Quand vous réalisez une prise d'origine de plusieurs axes de LinuxCNC, vous passez G53, les coordonnées système, à 0 pour chacun des axes concernés.

– La prises d'origine ne modifient en rien les autres systèmes de coordonnées, ni les compensations d'outil.

La seule façon de se déplacer en mode G53, en coordonnées machine, c'est de programmer un G53 sur la même ligne que celle d'un mouvement. En fonctionnement normal, vous êtes dans le système de coordonnées G54.

### 9.8.2 G54 à 59.3 Coordonnées utilisateur

Normalement vous utilisez le système de coordonnées G54. Quand un décalage est appliqué au système de coordonnées utilisateur courant, dans Axis, une petite sphère bleue avec des rayons est affichée à l'emplacement de l'origine machine quand la visu affiche *Position: Relative Actuelle*. Si votre décalage utilise temporairement les coordonnées machine, depuis le menu Machine ou en programmant *G10 L2 P1 X0 Y0 Z0* à la fin du programme G-Code. Modifiez la valeur du mot *P* en fonction du système de coordonnées dont vous voulez effacer le décalage.

– Les décalages stockés dans un système de coordonnées utilisateur sont conservés à l'arrêt de LinuxCNC.

– Dans Axis, utiliser le bouton *Toucher* décalera le système de coordonnées utilisateur choisi.

### 9.8.3 Quand vous êtes perdu

Si vous avez des difficultés pour obtenir 0,0,0 sur la visu alors que vous pensez que vous devriez l'avoir, c'est peut être provoqué par plusieurs décalages programmés et qu'il conviendrait de supprimer. Pour cela:

- Placez vous sur l'origine machine avec *G53 G0 X0 Y0 Z0*
- Supprimez tous les décalages *G92* avec *G92.1*
- Utilisez les coordonnées utilisateur avec *G54*
- Rendez les coordonnées utilisateur *G54*, identiques aux coordonnées machine avec *G10 L2 P1 X0 Y0 Z0 R0*
- Annulez les offsets d'outil avec *G49*
- Activez l'affichage des coordonnées relatives depuis le menu.

Maintenant vous devriez être, à l'origine machine *X0 Y0 Z0* et le système de coordonnées relatives devrait être le même que le système de coordonnées machine. :lang: fr :toc:

## Chapitre 10

# Aperçu global d'une machine CNC

Cette section donne une description des différents organes constituant une machine à commande numérique (CNC).

### 10.1 Composants mécaniques

Une machine à commande numérique dispose de beaucoup de composants mécaniques pouvant être contrôlés, ou qui peuvent avoir une incidence sur la façon dont le contrôle de la machine s'effectue. Cette section décrit les composants qui interagissent avec l'interpréteur. Les autres composants mécaniques, comme les boutons de jog, ne seront pas décrits ici, même si ils affectent le contrôle.

#### 10.1.1 Axes

Toute machine à commande numérique dispose d'un ou de plusieurs axes. Les différents types de machines ont différentes combinaisons d'axes. Par exemple, une fraiseuse 4 axes peut avoir la combinaison d'axes XYZA ou XYZB. Un tour classique aura les axes XZ. Une machine de découpe à fil chaud aura les axes XYUV.<sup>1 2</sup>

##### 10.1.1.1 Axes linéaires primaires

Les axes X, Y et Z produisent des mouvements linéaires dans trois directions, mutuellement orthogonales.

##### 10.1.1.2 Axes linéaires secondaires

Les axes U, V et W produisent des mouvements linéaires dans trois directions, mutuellement orthogonales. Habituellement, X et U sont parallèles, Y et V sont parallèles et Z et W sont parallèles.

##### 10.1.1.3 Axes rotatifs

Les axes A, B et C produisent des mouvements angulaires (rotations). Habituellement, l'axe de rotation de A est parallèle à X, l'axe de rotation de B est parallèle à Y et l'axe de rotation de C est parallèle à Z.

---

1. Si le mouvement des composants mécaniques n'est pas indépendant, comme sur une machine hexapode, le langage RS274/NGC et les fonctions standards seront quand même utilisables, tant que le contrôle de bas niveau sait comment contrôler les mécanismes actuels pour produire le mouvement relatif de l'outil et de la pièce qui auraient été produits par des axes indépendants. C'est appelé, la cinématique.

2. Avec LinuxCNC, le cas de la machine à portique XYYZ avec deux moteurs pour un axe est mieux traité par la cinématique que par un axe linéaire supplémentaire.

### 10.1.2 Broche

Une machine à commande numérique est équipée d'une broche qui maintient un outil coupant, un palpeur ou d'autres outils. La broche peut tourner dans les deux sens. Elle peut être conçue pour tourner à vitesse constante mais réglable. Excepté sur les machines dont la broche est montée sur un axe rotatif, l'axe de la broche est maintenu parallèle à l'axe Z et il est coïncident avec l'axe Z quand X et Y sont à zéro. La broche peut être stoppée sur une position fixée ou non.

### 10.1.3 Arrosages

Une machine à commande numérique peut être équipée d'un système fournissant l'arrosage fluide ou un arrosage par gouttelettes.

### 10.1.4 Correcteurs de vitesse d'avance et de broche

Une machine à commande numérique est équipée de boutons de réglage de la vitesse d'avance et de la vitesse de rotation de la broche, ils laissent l'opérateur corriger les vitesses nécessaires pour la broche et l'avance travail, il peut ainsi augmenter ou réduire les vitesses programmées.

### 10.1.5 Bouton d'effacement de bloc

Une machine à commande numérique peut être équipée d'un bouton d'effacement de bloc. Voir la section [effacement de bloc](#).

### 10.1.6 Bouton d'arrêt optionnel du programme

Une machine à commande numérique peut être équipée d'un bouton d'arrêt du programme. Voir la section [arrêts optionnels](#).

## 10.2 Composants de contrôle et de données

### 10.2.1 Axes linéaires

Les axes X, Y et Z forment un système de coordonnées orthogonales standard. La position d'un axe s'exprime en utilisant ses coordonnées.

### 10.2.2 Axes linéaires secondaires

Les axes U, V et W forment également un système de coordonnées standard. X et U sont parallèles, Y et V sont parallèles enfin Z et W sont parallèles.

### 10.2.3 Axes rotatifs

Les axes rotatifs se mesurent en degrés. Leur sens de rotation positif est le sens anti-horaire quand l'observateur est placé face à l'axe.<sup>3</sup>

### 10.2.4 Point contrôlé

Le point contrôlé est le point dont la position et la vitesse de déplacement sont contrôlés. Quand la compensation de longueur d'outil est à zéro (valeur par défaut), c'est un point situé sur l'axe de la broche et proche de la fin de celle-ci. Cette position peut être déplacée le long de l'axe de la broche en spécifiant une compensation de longueur d'outil. Cette compensation correspond généralement à la longueur de l'outil coupant courant. Ainsi, le point contrôlé est à la pointe de l'outil. Sur un tour, les correcteurs d'outil peuvent être spécifiés pour les axes X et Z, le point contrôlé est à la pointe de l'outil ou (correction du rayon de bec) légèrement en retrait du point d'intersection des droites perpendiculaires formées par l'axe des points de tangence à la pièce, de face et sur le côté de l'outil.

---

3. Si les parallélismes sont particuliers, le constructeur du système devra indiquer à quels sens de rotation correspondent horaire et anti-horaire.

### 10.2.5 Mouvement linéaire coordonné

Pour mener un outil sur une trajectoire spécifiée, une machine à commande numérique doit coordonner les mouvements de plusieurs axes. Nous utilisons le terme *mouvement linéaire coordonné* pour décrire une situation dans laquelle, nominalement, chacun des axes se déplace à vitesse constante et tous les axes se déplacent de leur point de départ à leur point d'arrivée en même temps. Si deux des axes X, Y, Z (ou les trois) se déplacent, ceci produit un mouvement en ligne droite, d'où le mot *linéaire* dans le terme. Dans les véritables mouvements, ce n'est souvent pas possible de maintenir la vitesse constante à cause des accélérations et décélérations nécessaires en début et fin de mouvement. C'est faisable, cependant, de contrôler les axes ainsi, chaque axe doit en permanence faire la même fraction du mouvement requis que les autres axes. Ceci déplace l'outil le long du même parcours et nous appelons aussi ce genre de mouvement, mouvement linéaire coordonné.

Un mouvement linéaire coordonné peut être exécuté soit en vitesse travail, soit en vitesse rapide, ou il peut être synchronisé à la rotation de la broche. Si les limites physiques de l'axe rendent le déplacement impossible, tous les axes seront ralentis pour maintenir le parcours prévu.

### 10.2.6 Vitesse d'avance

La vitesse à laquelle le point contrôlé se déplace est ajustable par l'opérateur. Sauf cas particulier, vitesse inverse du temps, vitesse par tour, voir la section [sur les modes de vitesse](#), dans l'interpréteur, l'interprétation des vitesses est la suivante:

1. Si le déplacement concerne un des axes XYZ, F est en unités machine par minute dans le système Cartésien XYZ et les mouvements des autres axes (UVWABC) sont également dans un même mode de coordonnées.
2. Autrement, si le déplacement concerne un des axes UVW, F est en unités machine par minute dans le système Cartésien UVW, tous les autres axes (ABC) se déplacent dans un même mode de coordonnées.
3. Autrement, le mouvement est purement rotatif et le mot F est en unités de rotation dans le système pseudo-Cartésien ABC.

### 10.2.7 Arrosage

Arrosage fluide ou par gouttelettes peuvent être activés séparément. Le langage RS274/NGC les arrête ensemble, voir la section [des contrôles d'arrosage](#).

### 10.2.8 Temporisation

Une temporisation peut être commandée (ex: pour immobiliser tous les axes) pendant une durée spécifique. La broche n'est pas arrêtée pendant une temporisation! Sans s'occuper [du mode de contrôle de trajectoire](#) la machine s'arrêtera exactement à la fin du dernier mouvement avant la temporisation.

### 10.2.9 Unités

Les unités utilisées pour les distances le long des axes X, Y et Z peuvent être les pouces ou les millimètres. La vitesse de rotation de la broche est en tours par minute. Les positions des axes rotatifs sont exprimées en degrés. Les vitesses d'avance sont exprimées en unités machine par minute ou en degrés par minute ou en unités de longueur par tour de broche, comme décrit dans la section [des vitesses](#).

### 10.2.10 Position courante

Le point contrôlé est toujours à un emplacement appelé la *position courante*, et le contrôleur sait toujours où est cette position. Les valeurs représentant la position courante doivent être ajustées en l'absence de tout mouvement des axes si un de ces événements a lieu:

1. Les unités de longueur ont changé.
2. La compensation de longueur d'outil a changé.
3. Le décalage d'origine a changé.

### 10.2.11 Choix du plan de travail

Il y a toujours un plan sélectionné, qui doit être le plan XY, le plan YZ, ou le plan XZ de la machine. L'axe Z est, bien sûr, perpendiculaire au plan XY, l'axe X perpendiculaire au plan YZ et l'axe Y perpendiculaire au plan XZ.

### 10.2.12 carrousel d'outils

Aucun ou un outil est assigné à chaque emplacement dans le carrousel.

### 10.2.13 Changeur d'outil

Une machine à commande numérique peut commander un changeur d'outils.

### 10.2.14 Chargeur de pièce

Les deux porte-pièces peuvent être intervertis par commande.

### 10.2.15 Chargeur de pièces

Une machine à commande numérique peut être équipée d'un système de chargement des pièces. Le système se compose de deux porte-pièces sur lesquels sont fixés les bruts des pièces à usiner. Un seul porte-pièce à la fois est en position d'usinage.

### 10.2.16 Boutons des correcteurs de vitesses

Les boutons des correcteurs de vitesses peuvent être activés (ils fonctionnent normalement) ou rendus inopérants (Ils n'ont plus aucun effet). Le langage RS274/NGC dispose d'une commande qui active tous les boutons et une autre qui les désactive. Voir l'inhibition et l'activation [des correcteurs de vitesse](#). Voir également [ici pour d'autres détails](#).

### 10.2.17 Modes de contrôle de trajectoire

La machine peut être placée dans un de ces trois modes de contrôle de trajectoire:

- mode arrêt exact:: En mode arrêt exact, le mobile s'arrête brièvement à la fin de chaque mouvement programmé.
- mode trajectoire exacte:: En mode trajectoire exacte, le mobile suit la trajectoire programmée aussi précisément que possible, ralentissant ou s'arrêtant si nécessaire aux angles vifs du parcours.
- mode trajectoire continue avec tolérance optionnelle:: En mode trajectoire continue, les angles vifs du parcours peuvent être légèrement arrondis pour que la vitesse soit maintenue (sans dépasser la tolérance, si elle est spécifiée).

Voir également les G-codes [G61/G61.1](#) et [G64](#) des contrôles de trajectoire.

## 10.3 Interaction de l'interpréteur avec les boutons

L'interpréteur interagit avec plusieurs boutons de commande. Cette section décrit ces interactions plus en détail. En aucun cas l'interpréteur ne connaît ce que sont les réglages de ces boutons.

### 10.3.1 Boutons de correction de vitesses

L'interpréteur de commande RS274/NGC autorise (M48) ou interdit (M49) l'action des boutons d'ajustement des vitesses. Pour certains mouvements, tels que la sortie de filet à la fin d'un cycle de filetage, les boutons sont neutralisés automatiquement.

LinuxCNC réagit aux réglages de ces boutons seulement quand ils sont autorisés.





## Chapitre 11

# Systèmes de coordonnées et décalages

### 11.1 Introduction

Dans ce chapitre, nous allons tenter de démystifier les systèmes de coordonnées. C'est une notion capitale pour maîtriser le fonctionnement d'une machine CNC, sa configuration et son utilisation.

Nous montrerons également, qu'il est très intéressant d'utiliser un point de référence sur le brut ou la pièce et de faire travailler le programme à partir de ce point, sans avoir à tenir compte d'où est placée la pièce sur la machine pendant l'usinage. Ce chapitre va donc introduire les décalages et comment ils sont utilisés par LinuxCNC:

- Les coordonnées machine.(G53)
- Les neuf décalages d'origine pièce.(G54 à G59.3)
- Un jeu de décalages globaux.(G92)

### 11.2 Commande en coordonnées machine: G53

Indépendamment de tout décalage pouvant être actif, un G53 dans une ligne de code indique à l'interpréteur de se déplacer aux positions réelles des axes (positions absolues), commandées dans la ligne. Par exemple:

```
G53 G0 X0 Y0 Z0
```

déplacera le mobile depuis la position actuelle vers la position où les coordonnées machine des trois axes seront à zéro. Vous pouvez utiliser cette commande si vous avez une position fixe pour le changement d'outil ou si votre machine dispose d'un changeur automatique. Vous pouvez aussi utiliser cette commande pour dégager la zone de travail et accéder à la pièce dans l'étau.

G53 est une commande non modale. Elle doit être utilisée sur chaque ligne où un mouvement basé sur les coordonnées machine est souhaité.









```
G1 F1 Z-0.25  
G3 X-0.1 Y0 I0.1 J0  
G54 G0 X0 Y0 Z0
```

```
M2
```

Maintenant c'est le moment d'appliquer une série de décalages G92 à ce programme. Vous verrez que c'est fait dans chaque cas de Z0. Si la machine était à la position zéro, un G92 Z1.0000 placé en tête de programme le décalerait d'un pouce. Vous pouvez également modifier l'ensemble du dessin dans le plan XY en ajoutant quelques décalages x et y avec G92. Si vous faites cela, vous devez ajouter une commande G92.1 juste avant le M2 qui termine le programme. Si vous ne le faites pas, les programmes que vous pourriez lancer après celui-ci, utiliseront également les décalages G92. En outre, cela permettrait d'éviter d'écrire les valeurs de G92 lorsque vous arrêtez LinuxCNC et donc, d'éviter de les recharger quand vous démarrez à nouveau le programme.

## Chapitre 12

# Les compensations d'outil

### 12.1 Compensation de longueur d'outil

#### 12.1.1 Toucher

Dans la boîte de dialogue du bouton *Toucher* de l'interface AXIS, il est possible de mettre à jour automatiquement la table d'outils.

Séquence typique pour mise à jour de la table d'outils:

- Après la prise d'origine, charger un outil *Tn M6* dans lequel *n* est le numéro de l'outil.
- Déplacer l'outil pour établir le zéro pièce, en utilisant une cale d'épaisseur ou en faisant une petite passe puis une mesure.
- Cliquer sur le bouton *Toucher* de l'onglet *Contrôle manuel* (ou presser la touche *Fin* du clavier).
- Sélectionner *Table d'outils* dans la liste déroulante des systèmes de coordonnées.
- Entrer l'épaisseur de la cale ou la cote mesurée.
- Presser OK.

La table d'outil sera alors modifiée avec la longueur correcte en Z de l'outil. La visu affichera la position en Z correcte et une commande G43 sera passée pour que la nouvelle longueur Z de l'outil soit effective. Le choix *Table d'outils* n'apparaîtra dans la liste déroulante du *Toucher*, que si l'outil à été chargé avec *Tn M6*.



FIGURE 12.1 – Toucher et table d'outils





## Chapitre 13

# Fichier d'outils et compensations

### 13.1 Fichier d'outils

Les longueurs et diamètres d'outils peuvent être lus [dans une table d'outils](#) ou provenir d'un mot spécifié pour activer la compensation d'outil.

### 13.2 Compensation d'outil

La compensation d'outil peut causer beaucoup de problèmes aux meilleurs programmeurs. Mais elle peut aussi être une aide puissante quand elle est utilisée pour aider l'opérateur à obtenir une pièce à la cote. En réglant la longueur et le diamètre des outils dans une table d'outils, les décalages peuvent être utilisés pendant un cycle d'usinage qui tient compte des variations de taille de l'outil, ou pour des déviations mineures des trajectoires programmées. Et ces changements peuvent être faits sans que l'opérateur n'ait à changer quoi que ce soit dans le programme.

Tout au long de ce module, vous trouverez occasionnellement des références à des fonctions canoniques, là où il est nécessaire pour le lecteur de comprendre comment fonctionne une compensation d'outil dans une situation spécifique. Ces références ont pour but de donner au lecteur une idée de la séquences plutôt que d'exiger qu'il comprenne la façon dont les fonctions canoniques elles-mêmes fonctionnent dans LinuxCNC.

### 13.3 Compensation de longueur d'outil

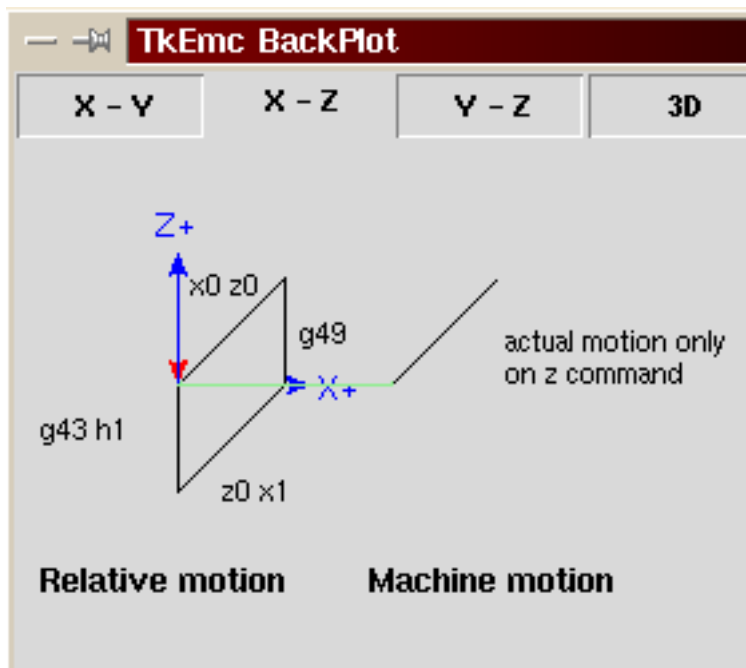
Les compensations de longueur d'outil sont données comme des nombres positifs dans la table d'outils. Une compensation d'outil est programmée en utilisant G43 Hn, où n est le numéro d'index de l'outil souhaité dans la table d'outil. Il est prévu que toutes les entrées dans la table d'outils soit positives. La valeur de H est vérifiée, elle doit être un entier non négatif quand elle est lue. L'interpréteur se comporte comme suit:

1. Si G43 Hn est programmé, un appel à la fonction `USE_TOOL_LENGTH_OFFSET(longueur)` est fait (où longueur est l'écart de longueur, lu dans la table d'outils, de l'outil indexé n), `tool_length_offset` est repositionné dans le modèle de réglages de la machine et la valeur de `current_z` dans le modèle est ajustée. Noter que n n'a pas besoin d'être le même que le numéro de slot de l'outil actuellement dans la broche.
2. Si G49 est programmé, `USE_TOOL_LENGTH_OFFSET(0.0)` est appelée, `tool_length_offset` est remis à 0.0 dans le modèle de réglages de la machine et la valeur courante de `current_z` dans le modèle est ajustée. L'effet de la compensation de longueur d'outil est illustrée dans la capture ci-dessous. Noter que la longueur de l'outil est soustraite de Z pour que le point contrôlé programmé corresponde à la pointe de l'outil. Il faut également noter que l'effet de la compensation de longueur est immédiat quand on voit la position de Z comme une coordonnée relative mais il est sans effet sur la position actuelle de la machine jusqu'à ce qu'un mouvement en Z soit programmé.

Programme de test de longueur d'outil.

L'outil #1 fait un pouce de long.

```
N01 G1 F15 X0 Y0 Z0
N02 G43 H1 Z0 X1
N03 G49 X0 Z0
N04 G0 X2
N05 G1 G43 H1 G4 P10 Z0 X3
N06 G49 X2 Z0
N07 G0 X0
```



Avec ce programme, dans la plupart des cas, la machine va appliquer le décalage sous forme d'une rampe pendant le mouvement en xyz suivant le mot G43.

## 13.4 Compensation de rayon d'outil

La compensation de rayon d'outil permet de suivre un parcours sans se préoccuper du diamètre de l'outil. La seule restriction, c'est que les *mouvements d'entrée* doivent être au moins aussi long que le rayon de l'outil utilisé.

Il y a deux parcours que l'outil peut prendre pour usiner un profil quand la compensation de rayon est activée, un parcours à gauche du profil et un à droite du profil. Pour les visualiser, il faut s'imaginer être debout sur la pièce, marchant en suivant l'outil pendant que celui-ci progresse dans la matière. G41 fait passer l'outil à gauche du profil et G42 le fait passer à droite du profil.

Le point final de chaque mouvement, dépend du mouvement suivant. Si le mouvement suivant crée un angle extérieur, le mouvement se terminera à l'extrémité de la ligne de coupe compensée. Si le mouvement suivant crée un angle intérieur, l'outil s'arrêtera avant d'interférer avec la matière de la pièce. La figure suivante montre comment le mouvement se termine à différents endroits, dépendants du mouvement suivant.

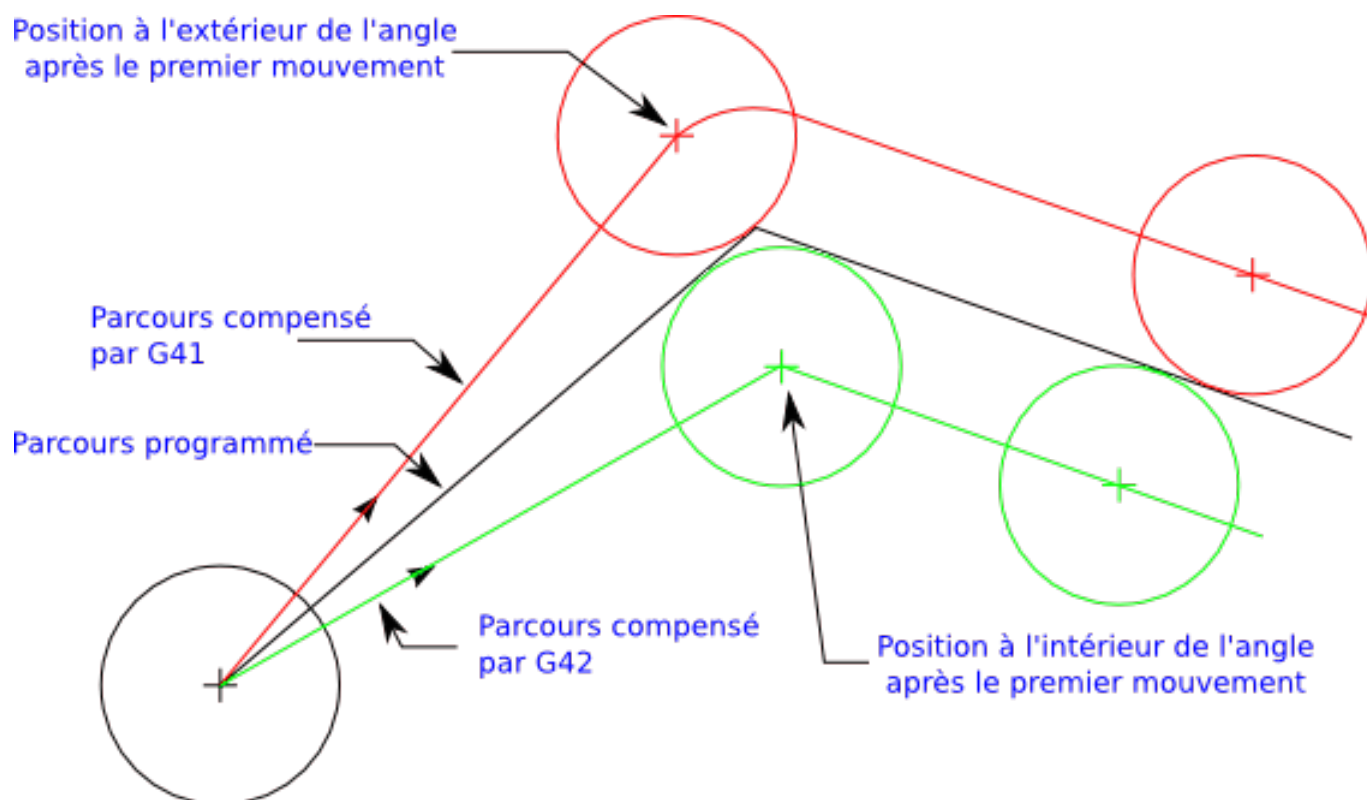


FIGURE 13.1 – Point final de la compensation

### 13.4.1 Vue générale

#### 13.4.1.1 Table d'outils

La compensation de rayon d'outil utilise les données de la table d'outils pour déterminer le décalage nécessaire. Les données peuvent être introduites à la volée, avec G10 L1.

#### 13.4.1.2 Programmation des mouvements d'entrée

Tout mouvement suffisamment long pour arriver en position compensée, sera un mouvement d'entrée valide. La longueur minimale équivaut au rayon de l'outil. Ça peut être un mouvement en vitesse rapide au dessus de la pièce. Si plusieurs mouvements en vitesse rapide sont prévus après un G41/G42, seul le dernier placera l'outil en position compensée.

Dans la figure suivante, on voit que le mouvement d'entrée est compensé à droite du profil. Ce qui aura pour effet, lors du mouvement d'entrée, de déplacer le centre de l'outil, d'un rayon d'outil à droite de X0. Dans ce cas, le mouvement d'entrée laissera un petit plot de matière en raison du décalage de compensation et de l'arrondi de l'outil.

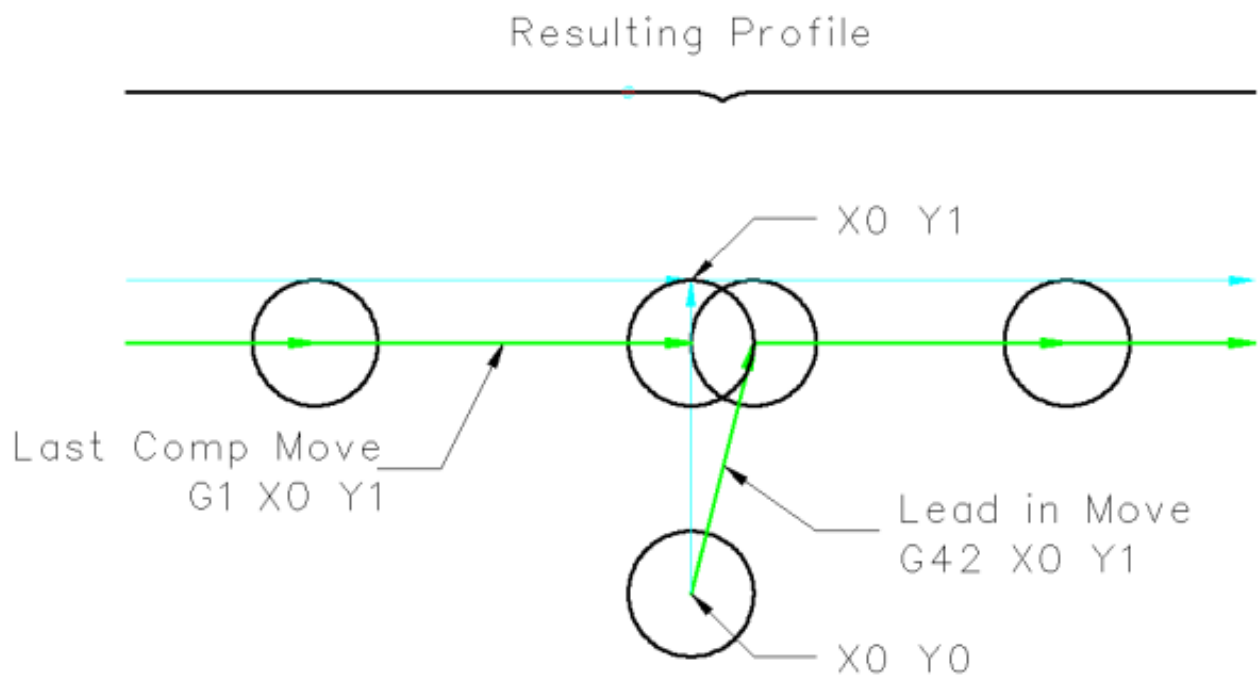


FIGURE 13.2 – Mouvement d'entrée

#### 13.4.1.3 Mouvement en Z

Un mouvement en Z est possible pendant que le contour est suivi dans le plan XY. Des portions du contour peuvent être sautées en rétractant l'axe Z au dessus du bloc et en amenant Z au dessus du prochain point de départ.

#### 13.4.1.4 Mouvement en vitesse rapide

Des mouvements en vitesse rapide peuvent être programmé avec les compensations d'outil actives.

#### 13.4.1.5 Bonne pratique

- Débuter tout programme avec un G40 pour être sûr que la compensation est désactivée.

## 13.4.2 Exemples de profils

### 13.4.2.1 Profil extérieur

#### G-Code

F25 (Vitesse d'avance travail)

G40 (Révocation des compensations)

G10 L1 P1 R.25 Z1 (Réglages en table d'outils)

T1 M6 (Appel de l'outil 1)

G42 (Compensation d'outil à droite du profil)

G1 X1 Y1 (Mouvement d'entrée)

X3 (Parcours trajectoire de coupe)

Y3

X1

Y1

G40 (Révocation de la compensation)

G0 Y0 Y0 (Dégagement de l'outil)

M2 (Fin de programme)

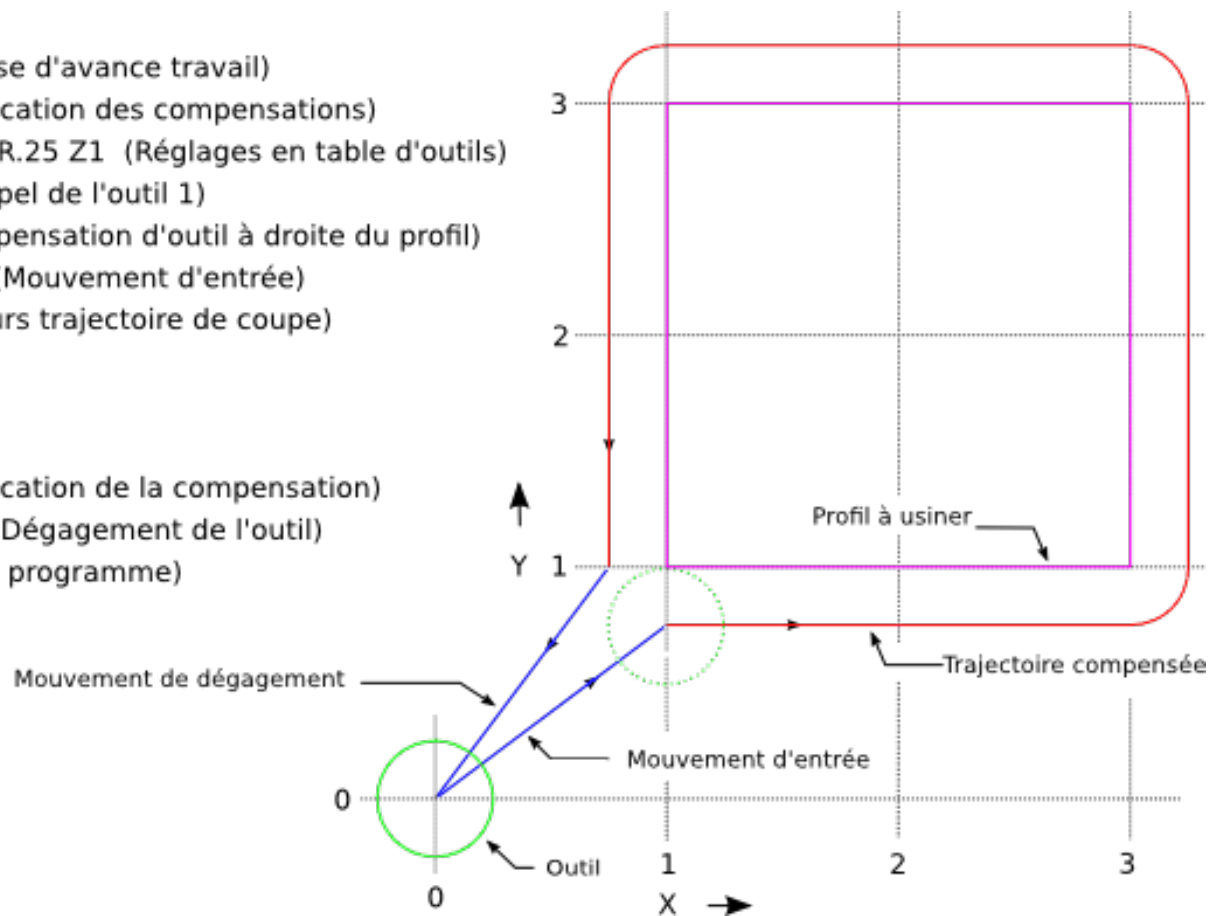


FIGURE 13.3 – Profil extérieur



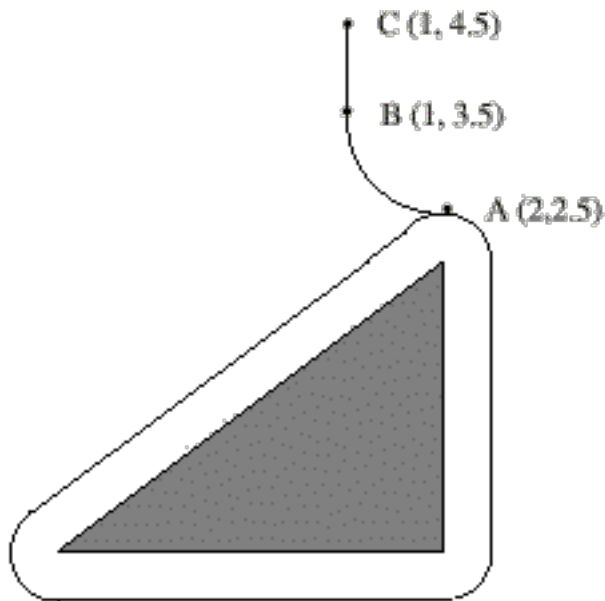












**Figure 11. Cutter Radius Compensation Entry Moves**

Dans cet exemple, l'arc AB et la ligne BC sont très larges, ce n'est pas nécessaire. Pour un contour sur parcours d'outil, le rayon de l'arc AB demande juste à être légèrement plus grand que la variation maximale du rayon de l'outil par rapport à son rayon nominal. Également, pour un contour sur parcours d'outil, le côté choisi pour la compensation doit être celui utilisé si l'outil est sur-dimensionné. Comme mentionné précédemment, si l'outil est sous-dimensionné, l'interpréteur basculera de l'autre côté.

### 13.5.2.3 Méthode simple

Si le contour est sur le profil du matériau et qu'il comprends des angles aigus quelque part sur le contour, une méthode simple pour faire l'entrée est possible. Voir la figure 12.

Premièrement, choisir un angle aigu, par exemple D. Ensuite, décider comment on va tourner autour du matériau depuis le point D. Dans notre exemple nous maintiendrons l'outil à gauche du profil et nous avancerons vers F. Prolonger la ligne FD (si le segment suivant du contour est un arc, prolonger la tangente à l'arc FD depuis D) pour diviser la surface extérieure au contour proche de D en deux parties. S'assurer que le centre de l'outil est actuellement dans la partie du même côté de la ligne prolongée que le matériau. Sinon, déplacer l'outil dans cette partie. Par exemple, le point E représente la position courante du centre de l'outil. Comme il est du même côté de la ligne FD prolongée que le triangle gris du matériau, aucun mouvement supplémentaire n'est nécessaire. Maintenant écrire la ligne de code NC qui active la compensation et faire le mouvement vers le point D

```
N0010 G41 G1 X2 Y2 (active la compensation et fait le mouvement d'entrée)
```

Cette méthode fonctionnera également avec un angle aigu sur un contour sur parcours d'outil, si l'outil est sur-dimensionné, mais elle échouera si il est sous-dimensionné.





supprime tous ce désordre et il est beaucoup plus facile de travailler avec lui, de toute façon. Si l'arc est de plus de 180 degrés, R doit être négatif.



































































































































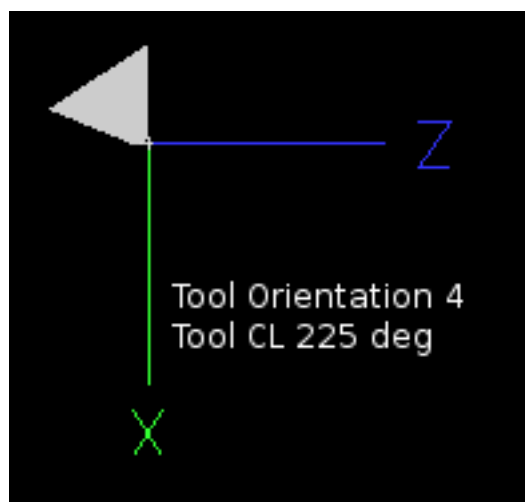
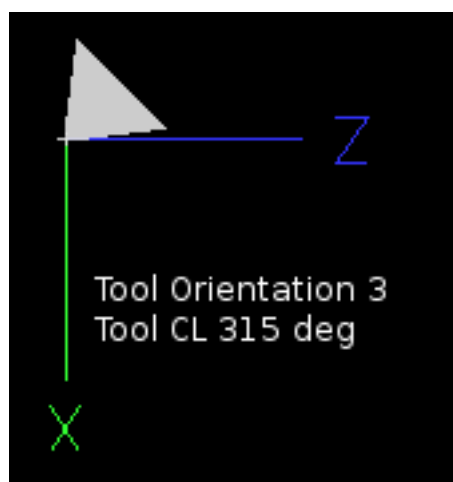
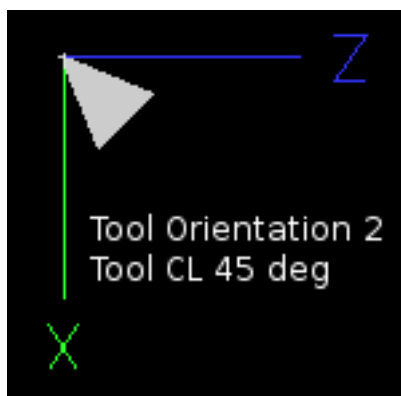




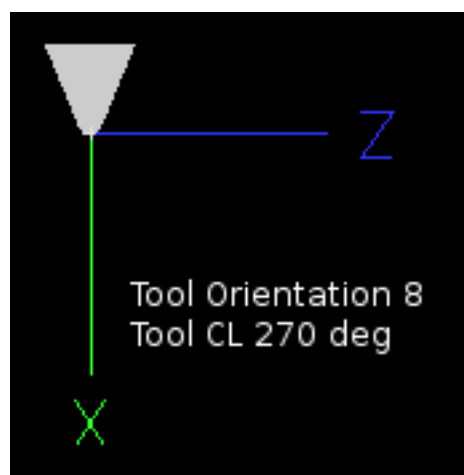
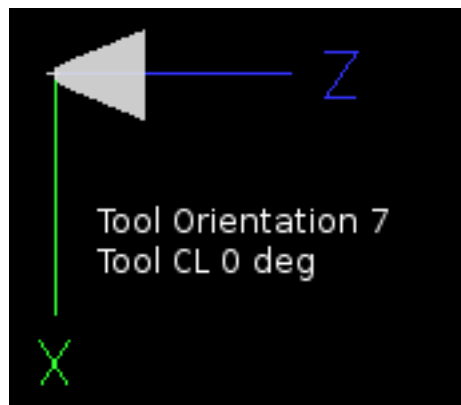
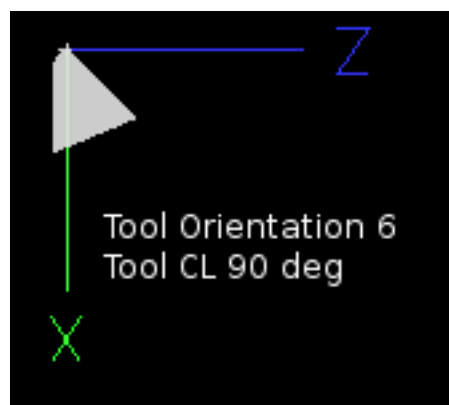
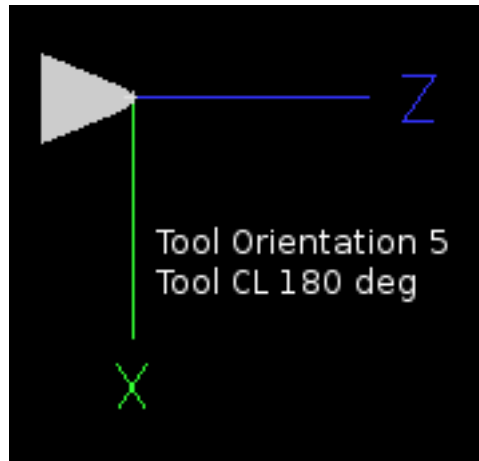








Outil dans les positions 5, 6, 7 et 8



## 20.4 Correction d'outil

Quand AXIS est utilisé sur un tour, il est possible de corriger l'outil sur les axes X et Z. Les corrections sont alors introduites directement dans la table d'outils en utilisant le bouton *Toucher* et sa fenêtre de dialogue.

### 20.4.1 Offset d'outil en X

L'offset X pour chaque outil correspond à un décalage de l'axe de la broche.

Une méthode consiste à prendre un outil de tournage standard et usiner un diamètre. Mesurer exactement ce diamètre puis, sans toucher à l'axe X, dans la fenêtre qui s'ouvre, après un appui sur le bouton *Toucher*, saisir le diamètre mesuré, ou le rayon si c'est le mode en cours. Ensuite, à l'aide d'encre à tracer ou d'un marqueur, recouvrir une zone sur la pièce, faire tangenter l'outil à cet endroit pour qu'il touche juste la surface encrée, ajuster alors l'offset X au diamètre mesuré de la pièce en utilisant le bouton *Toucher*. S'assurer que le rayon de bec est bien défini dans la table d'outils, pour que le point contrôlé soit correct. Le *Toucher* ajoute automatiquement un G43, de sorte que l'offset s'applique immédiatement à l'outil courant.

### 20.4.2 Séquence typique de correction d'outil en X:

1. Prise d'origine machine de chacun des axes, si ce n'est pas déjà fait.
2. Déclarer l'outil avec *M6 Tn* dans lequel *n* est le numéro de l'outil courant, présent en table d'outils.
3. Sélectionner l'axe X dans la fenêtre de l'onglet *Contrôle manuel (F3)*.
4. Déplacer l'axe X sur une position connue ou prendre une passe de test puis mesurer le diamètre obtenu.
5. Cliquer le bouton *Toucher* et choisir l'option *Table d'outils*, ce qui entrera la correction directement dans la table d'outil.
6. Recommencer la même séquence pour corriger l'axe Z.

Remarque: si le mode rayon est le mode courant, il faut évidemment entrer le rayon et non pas le diamètre.

### 20.4.3 Offset d'outil en Z

L'offset de l'axe Z peut être un peu déroutant au premier abord car il est composé de deux éléments. Le premier est l'offset de la table d'outils, le second est l'offset des coordonnées machine. Nous allons d'abord examiner l'offset de la table d'outils. Une méthode consiste à utiliser un point fixe sur le tour et à ajuster l'offset Z de tous les outils à partir de ce point fixe. Certains utilisent le nez de broche ou la face du mandrin. Cela donne la possibilité de changer d'outil et d'ajuster son offset Z, sans avoir à réinitialiser tous les outils.

### 20.4.4 Séquence typique de correction d'outil en Z:

1. Prise d'origine machine de tous les axes, si ce n'est pas déjà fait.
2. S'assurer qu'aucune compensation n'est activée pour le système de coordonnées courant.
3. Déclarer l'outil avec *M6 Tn* dans lequel *n* est le numéro de l'outil courant, présent en table d'outils.
4. Sélectionner l'axe Z dans la fenêtre de l'onglet *contrôle manuel (F3)*.
5. Placer un cylindre dans le mandrin.
6. Faire tangenter l'outil contre la face du cylindre.
7. Cliquer le bouton *Toucher* puis choisir *Table d'outils* et saisir la position à 0.0.
8. Répéter l'opération pour chaque outil, en utilisant le même cylindre.

Maintenant, tous les outils sont compensés à la même distance d'une position standard. Si un outil doit être changé, par exemple par un foret il suffira de répéter la séquence précédente pour qu'il soit synchronisé avec l'offset Z du reste des outils. Certains outils pourraient nécessiter un peu de réflexion pour déterminer le point contrôlé par rapport au point de *Toucher*. Par exemple, un outil de tronçonnage de 3.17mm d'épaisseur qui est touché sur le côté gauche, alors que l'opérateur veut Z0 sur le côté droit, il lui faudra alors saisir 3.17 dans la fenêtre du *Toucher*.

### 20.4.5 Machine avec tous les outils compensés

Une fois que tous les outils ont leurs offsets renseignés dans la table d'outils, il est possible d'utiliser n'importe quel outil présent en table d'outils pour ajuster le décalage du système de coordonnées machine.

### 20.4.6 Séquence typique de décalage du système de coordonnées:

1. Prise d'origine machine de tous les axes, si ce n'est pas déjà fait.
2. Déclarer l'outil avec *M6 Tn* dans lequel *n* est le numéro de l'outil courant, présent en table d'outils.
3. Envoyer un G43 pour que l'offset de l'outil soit activé. (voir ci-dessous)
4. Tangenter l'outil contre la pièce et fixer l'offset machine Z.

Ne pas oublier d'envoyer le G43 sur l'outil avant de définir le décalage du système de coordonnées machine, les résultats ne seraient pas ceux attendus... puisque la compensation de l'outil serait ajoutée à l'offset courant lorsque l'outil sera utilisé dans le programme.

## 20.5 Mouvements avec broche synchronisée

Sur un tour, les mouvements avec broche synchronisée nécessitent un signal de retour entre la broche et LinuxCNC. Généralement, c'est un codeur en quadrature qui fournit ce retour. Le manuel de l'intégrateur donne des explications sur l'utilisation des codeurs de broche.

**Filetage** Le cycle de filetage préprogrammé G76 est utilisé, tant en filetage intérieur qu'en filetage extérieur, voir [la section G76](#).

**Vitesse de coupe à surface constante** La vitesse de coupe à surface constante utilise l'origine machine X modifiée par l'offset d'outil X, pour calculer la vitesse de rotation de la broche en tr/mn. La vitesse de coupe à surface constante permet de suivre les changements d'offset de l'outil. L'emplacement de l'origine machine de l'axe X doit être sur l'axe de rotation et doit se faire avec l'outil de référence (celui qui a l'offset à zéro).

**Avance par tour** L'avance par tour déplace l'axe Z de la valeur de F à chaque tour. Ce n'est pas destiné au filetage pour lequel il faut utiliser G76. D'autres informations sont dans la section sur [G95](#).

## 20.6 Arcs

Le calcul des arcs peut être un exercice assez compliqué, même sur un tour, sans considérer les modes rayon et diamètre, ni l'orientation du système de coordonnées machine. Ce qui suit s'applique à des arcs au format centre. Sur un tour, il faut inclure G18 dans le préambule du programme G-code pour remplacer le G17 par défaut, le fait d'être en mode tour dans Axis ne suffit pas. Les arcs en G18, plan XZ utilisent les offsets pour I (l'axe X) et K (l'axe Z).

### 20.6.1 Les arcs et la cinématique du tour

Le tour classique a la broche à gauche de l'opérateur et l'outil entre l'opérateur et le centre de rotation du mandrin. C'est un agencement avec un axe Y(+) imaginaire pointant vers le sol.

Ce qui suit est valable pour ce type d'agencement:

- Le côté positif de l'axe Z pointe vers la droite, en s'éloignant de la broche.
- Le côté positif de l'axe X pointe vers l'opérateur, quand il est du côté de l'opérateur par rapport au centre de rotation, ses valeurs sont positives.

Certains tours ont l'outil du côté arrière et un axe Y(+) imaginaire pointant vers le haut.

Les directions des arcs G2/G3 sont basées sur l'axe autour duquel ils tournent. Dans le cas des tours, il s'agit de l'axe imaginaire Y. Si l'axe Y(+) pointe vers le sol, il faut regarder vers le haut pour que l'arc paraisse aller dans la bonne direction. Alors qu'en regardant depuis le dessus il faut inverser les G2/G3 pour que l'arc semble aller dans la bonne direction.

## 20.6.2 Mode rayon et mode diamètre

Lors du calcul des arcs en mode rayon, il suffit de se rappeler la direction de rotation telle qu'elle s'applique à ce tour.

Lors du calcul des arcs en mode diamètre, X est le diamètre, l'offset X (I) est le rayon, même en mode diamètre G7.

## 20.7 Parcours d'outil

### 20.7.1 Point contrôlé

Le point contrôlé pour l'outil, suit la trajectoire programmée. Le point contrôlé est l'intersection entre deux lignes parallèles aux axes X et Z, tangentes au rayon de bec de l'outil, définies en faisant tangenter l'outil en X puis en Z. En cylindrage ou en dressage de face sur une pièce, la trajectoire de coupe et l'arête de coupe de l'outil suivent le même parcours. Lors du tournage d'un rayon ou d'un angle, l'arête de coupe de l'outil ne suit pas la trajectoire programmée, sauf si la compensation d'outil est activée. Dans la figure suivante, on voit bien que le point contrôlé n'est pas sur l'arête de coupe de l'outil comme on pourrait le supposer.

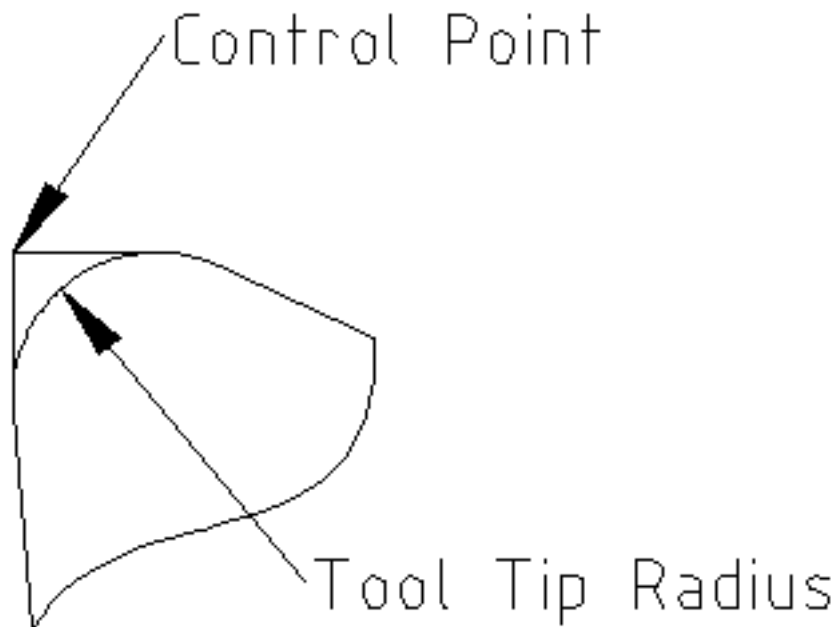


FIGURE 20.2 – Point contrôlé

### 20.7.2 Tourner les angles sans compensation d'outil

Maintenant imaginons de programmer une rampe sans compensation d'outil. La trajectoire programmée est représentée sur la figure suivante. Comme on peut le voir, la trajectoire programmée et la trajectoire de coupe souhaitée sont identiques uniquement si les mouvements de tournage suivent les axes X et Z.

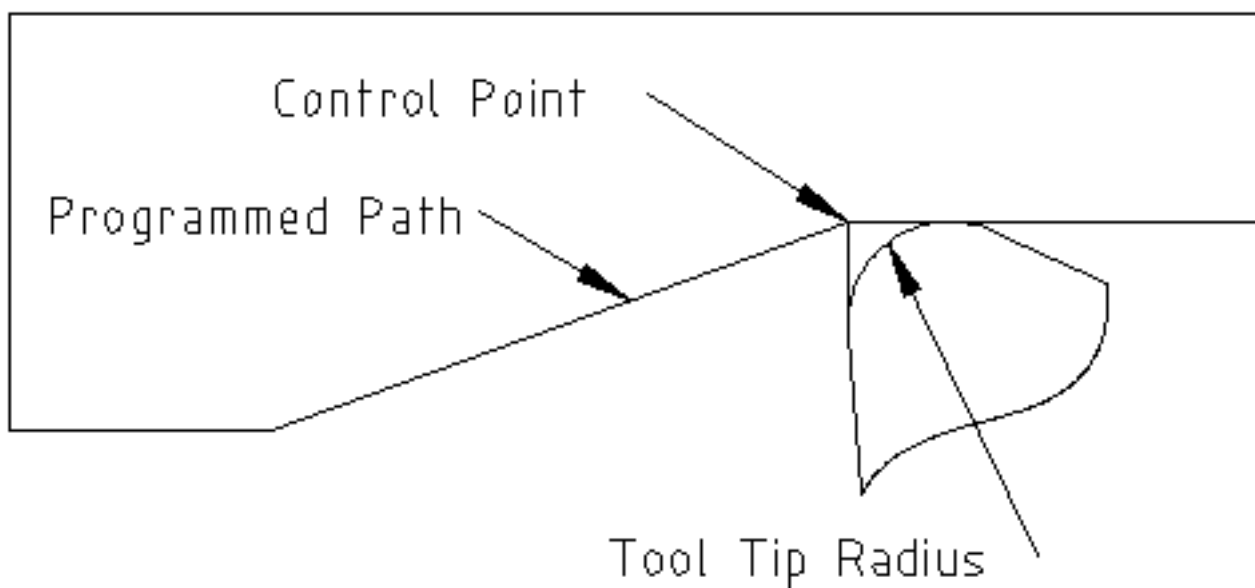


FIGURE 20.3 – Tournage en rampe

Le point contrôlé progresse en suivant la trajectoire programmée mais l'arête de coupe ne suit pas cette trajectoire comme c'est visible sur la figure suivante. Pour résoudre ce problème, il est nécessaire d'activer la compensation d'outil et d'ajuster la trajectoire programmée pour compenser le rayon de bec de l'outil.

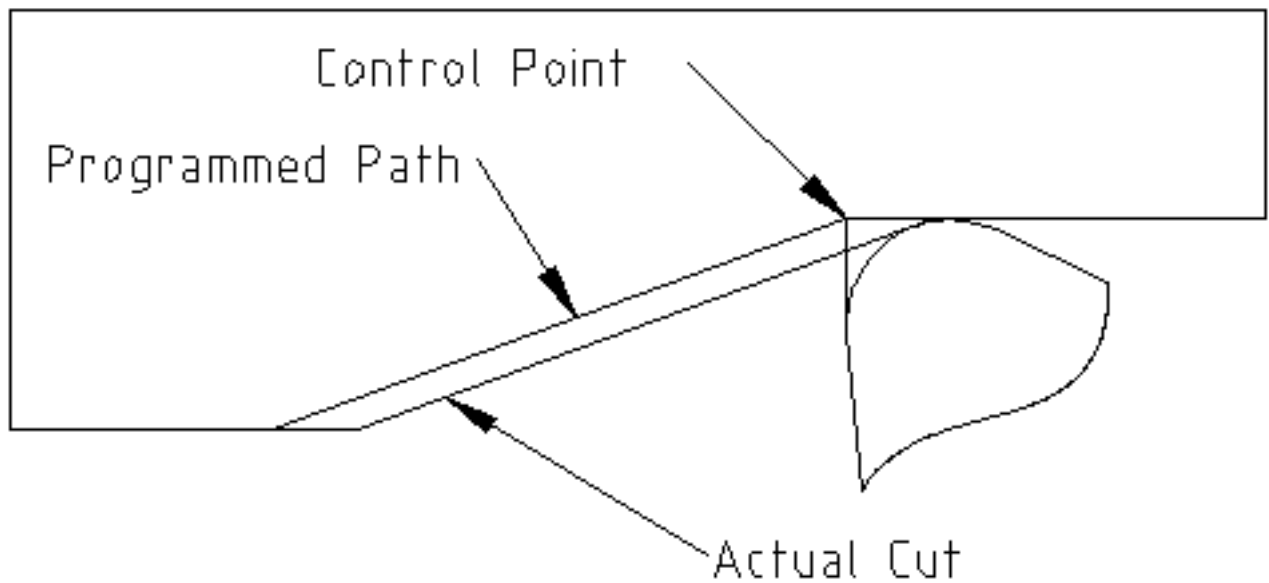


FIGURE 20.4 – Trajectoire en rampe

Dans l'exemple ci-dessus, pour suivre la rampe programmée et obtenir la bonne trajectoire, il suffit de décaler la trajectoire de la rampe vers la gauche, de la valeur d'un rayon de bec.

### 20.7.3 Tournage avec rayon extérieur

Dans cet exemple nous allons examiner ce qui se passe durant le tournage d'un rayon extérieur sans compensation de rayon de bec. Sur la figure suivante on voit l'outil tourner un diamètre extérieur sur la pièce. Le point contrôlé de l'outil suit bien la trajectoire programmée, l'outil touche le diamètre extérieur de la pièce.

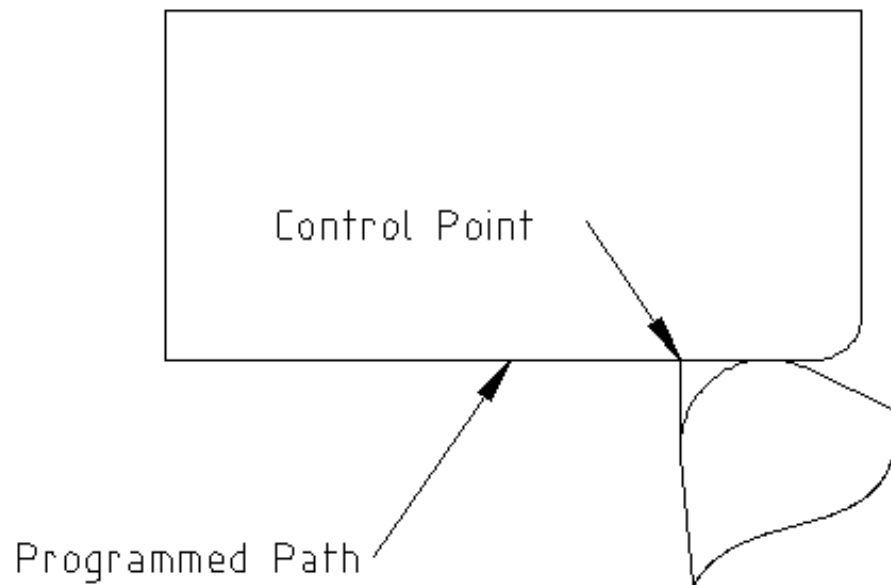


FIGURE 20.5 – Tournage du diamètre

Sur la figure suivante, on voit que quand l'outil approche la fin la pièce, le point contrôlé continue de suivre la trajectoire alors que l'arête de coupe a déjà quitté la matière et coupe en l'air. On voit aussi que malgré qu'un rayon a été programmé, la pièce conserve son angle d'extrémité.

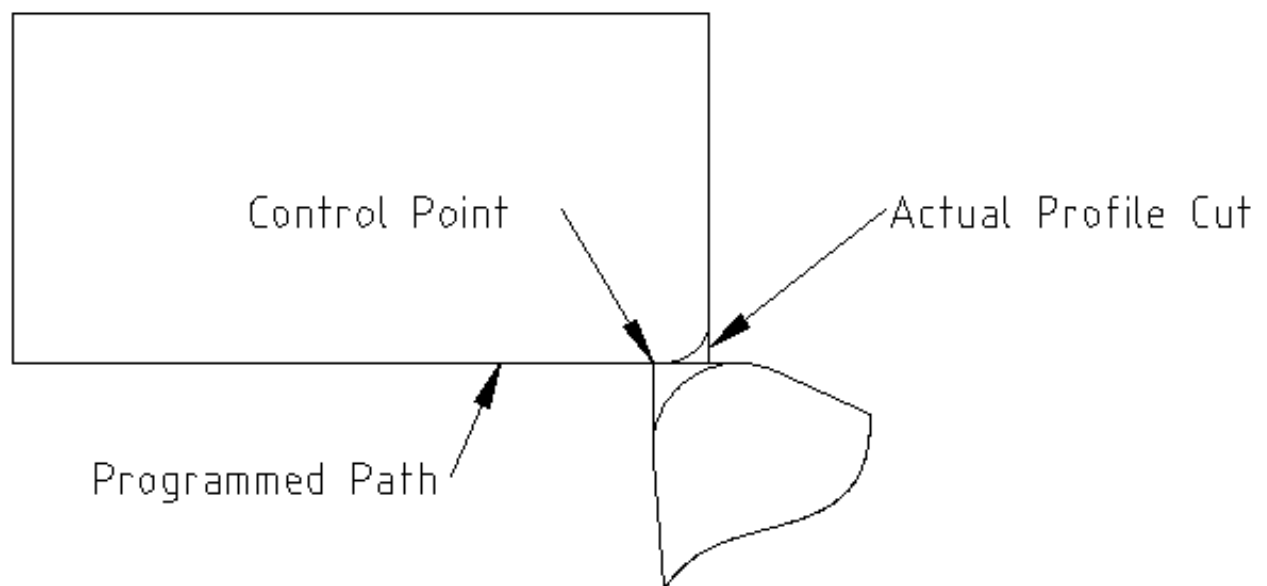


FIGURE 20.6 – Tournage du rayon

Maintenant, comme on le voit, le point contrôlé suit bien la trajectoire programmée mais l'arête de coupe est en dehors de la matière.

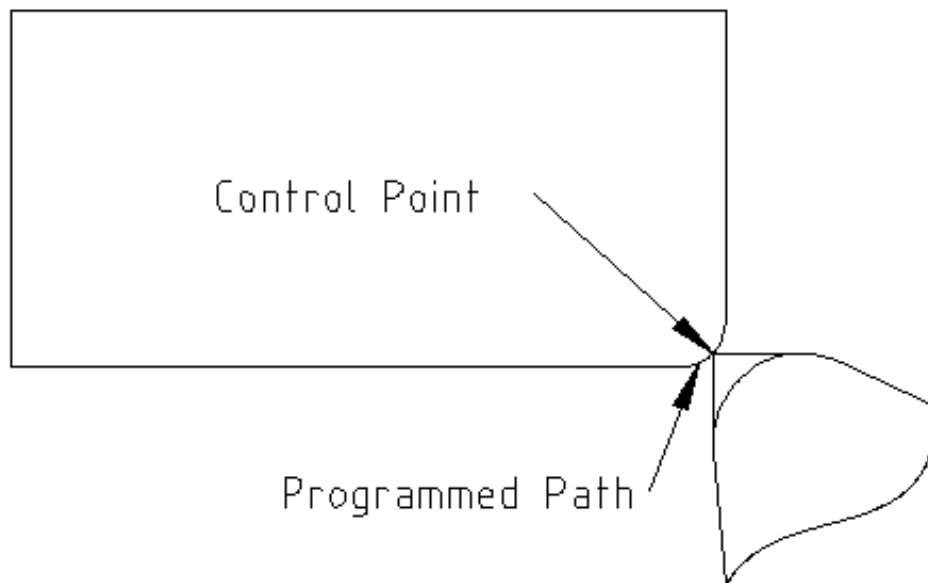


FIGURE 20.7 – Tournage du rayon

Sur la figure finale, on voit que l'arête de coupe a terminé le dressage de la face mais en laissant un coin carré à la place du beau rayon attendu. Noter aussi que, pour la même raison, pour ne pas laisser de téton au centre de la pièce lors du dressage de sa face, il convient de dépasser le centre de rotation de la valeur d'un rayon de bec de l'outil.

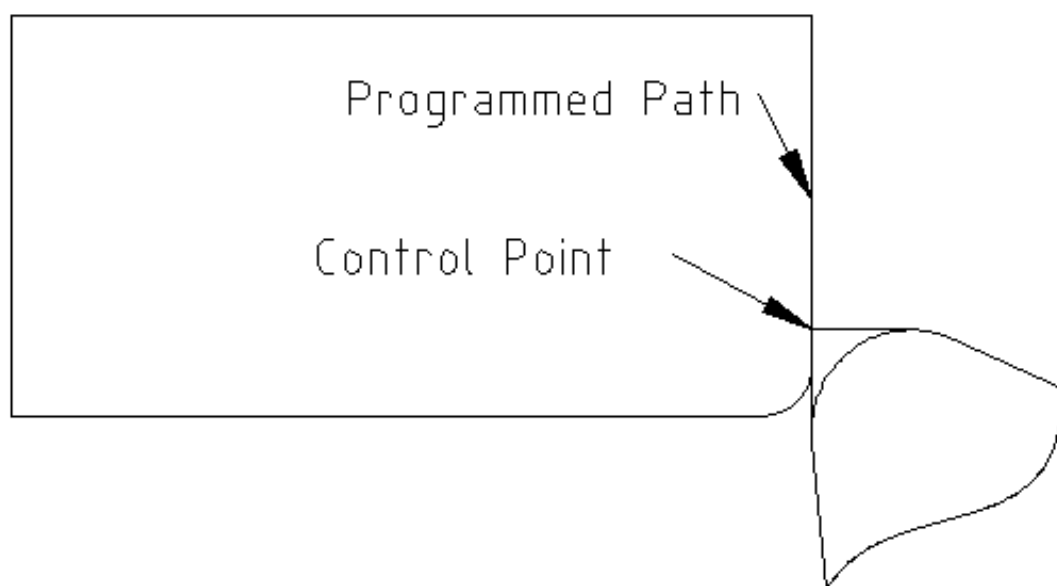


FIGURE 20.8 – Dressage de la face

#### 20.7.4 Utiliser la compensation d'outil

- Quand la compensation d'outil est utilisée sur un tour, penser à l'arête de coupe de l'outil comme étant celle d'un outil rond.
- Quand la compensation d'outil est utilisée, la trajectoire doit être suffisamment large pour qu'un outil rond n'interfère pas avec la pièce à la ligne suivante.

- Pour tourner des lignes droites sur un tour, il est préférable de ne pas utiliser la compensation d'outil. Par exemple pour aléser un trou avec une barre d'alésage un peu grosse, la place pourrait manquer pour dégager l'outil et faire le mouvement de sortie.
- Le mouvement d'entrée dans un arc avec la compensation d'outil, est important pour obtenir des résultats corrects.

## Chapitre 21

# Différences avec RS274/NGC

### 21.1 Changements entre RS274/NGC et LinuxCNC

#### 21.1.1 Position après un changement d'outil

Avec LinuxCNC, le mobile ne retourne pas sur la position de départ après un changement d'outil. Ce mode de fonctionnement est nécessaire, car un outil peut être plus long que l'outil précédent et dans ce cas un mouvement sur la position précédente placerait l'outil trop bas.

#### 21.1.2 Les paramètres de décalage sont dans l'unité du fichier ini

Dans LinuxCNC, les valeurs mémorisées dans les paramètres pour les positions d'origine des commandes G28 et G30, les systèmes de coordonnées P1 à P9 et le décalage G92 sont dans l'unité du fichier ini. Ce changement a été fait car la position d'un point change selon que G20 ou G21 était actif lors de la programmation d'un G28, G30, G10 L2 ou G92.3.

#### 21.1.3 Table d'outils, longueur et diamètre sont dans l'unité du fichier ini

Dans LinuxCNC, les longueurs d'outil (compensation) et les diamètres sont spécifiés seulement dans l'unité du fichier ini. Cela est nécessaire, car la longueur et le diamètre de l'outil changent selon que G20 ou G21 étaient actifs à l'initialisation des modes G43, G41, G42. Il était donc impossible de lancer un G-code avec des unités machines non natives, ceci même lorsque le G-code est simple et bien formé (débutant par G20 ou G21 et sans changement d'unité tout au long du programme) sans changer la table d'outils.

#### 21.1.4 G84, G87 ne sont pas implémentés

G84 et G87 ne sont pour le moment pas implémentés. Ils le seront dans une version futur de LinuxCNC.

#### 21.1.5 G28, G30 avec des mots d'axe

Lorsqu'un G28 ou un G30 est programmé avec seulement quelques mots d'axe présents, LinuxCNC déplace seulement les axes nommés. Ce comportement est commun aux contrôleurs de machine. Pour déplacer certains axes à un point intermédiaire, puis déplacer tous les axes à un point prédéfini, écrire deux lignes de G-code:

```
G0 X- Y- (édplace les axes au point éintermdiaire)
G28      (édplace tous les axes au point ééprdfini)
```

## 21.2 Ajouts à RS274/NGC

Différences qui ne changent pas le déroulement des programmes en RS274/NGC.

### 21.2.1 Codes de filetage G33 et G76

Ces codes ne sont pas définis dans RS274/NGC.

### 21.2.2 G38.2

La pointe de touche n'est pas rétractée après un mouvement G38.2. Ce mouvement de retrait sera ajouté dans une version futur de LinuxCNC.

### 21.2.3 G38.3 à G38.5

Ces codes ne sont pas définis dans RS274/NGC.

### 21.2.4 Les O-codes

Ces codes ne sont pas définis dans RS274/NGC

### 21.2.5 M50 à M53 Correcteurs de vitesse

Ces codes ne sont pas définis dans RS274/NGC.

### 21.2.6 M61 à M66

Ces codes ne sont pas définis dans RS274/NGC.

### 21.2.7 G43, G43.1

#### Longueurs d'outil négatives

Les spécifications RS274/NGC précisent "il est prévu que" toutes les longueurs d'outils soient positives. Cependant, G43 fonctionne avec des longueurs d'outil négatives.

#### Outils de tournage

La compensation de longueur d'outil G43 peut compenser l'outil à la fois en X et en Z. Cette fonctionnalité est surtout utile sur les tours.

#### Longueurs d'Outil dynamiques

LinuxCNC permet la spécification d'une longueur d'outil calculée par G43.1 I K.

### 21.2.8 G41.1, G42.1 Compensation dynamique

LinuxCNC permet dans le G-code, la spécification d'un diamètre d'outil et en mode tour, l'orientation est également spécifiée. Le format est G41.1/G42.1 D L, où D est le diamètre et L (si spécifié) est l'orientation de l'outil de tournage.

### 21.2.9 G43 sans le mot H

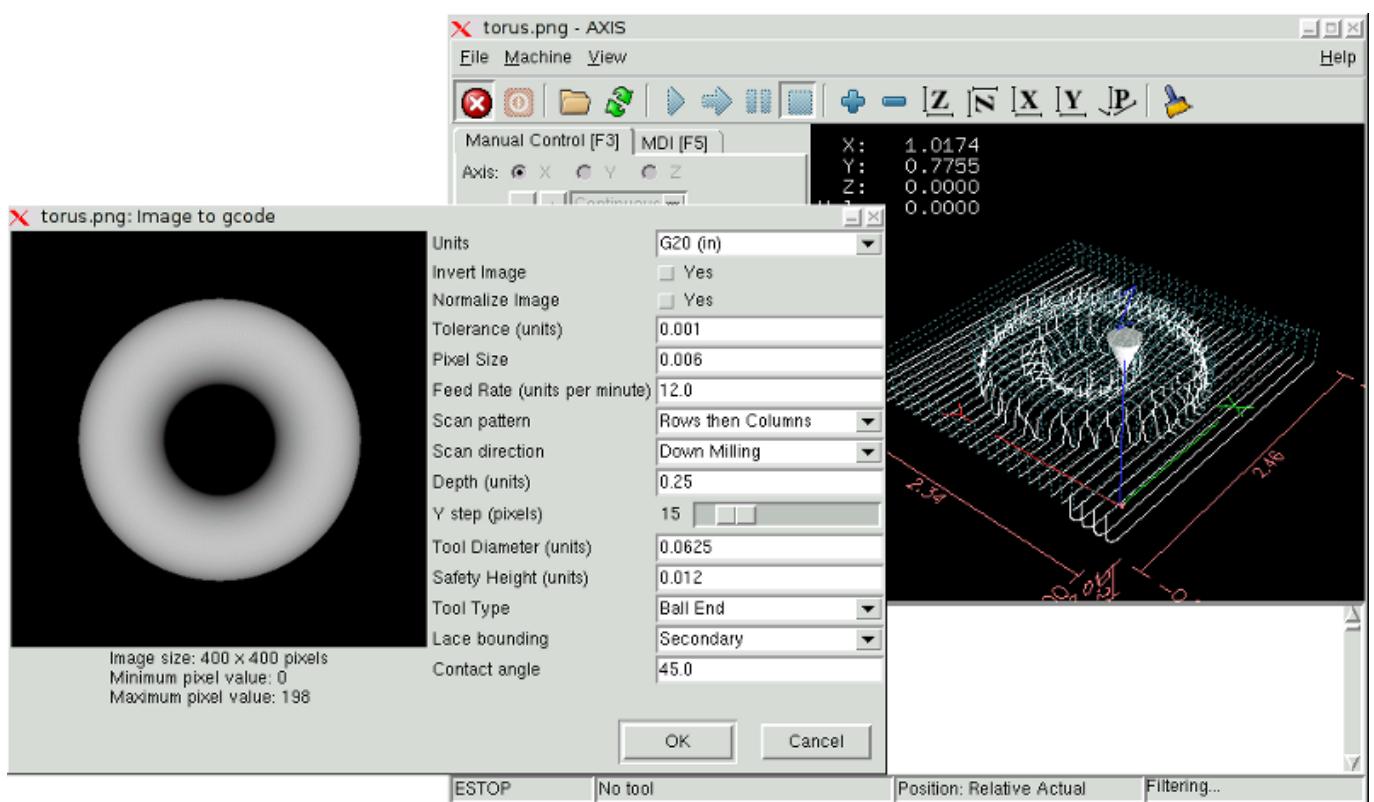
Ce code n'est pas permis en NGC. Dans LinuxCNC, il fixe la compensation de longueur pour l'outil actuellement chargé. Si aucun outil n'est actuellement chargé, c'est une erreur. Ceci a été fait afin que l'utilisateur n'ait pas à spécifier, pour chaque changement d'outil, le numéro d'outil à deux endroits et c'est cohérent avec la manière de fonctionner de G41/G42 quand le mot D n'est pas spécifié.

### **21.2.10 U, V et W axes**

LinuxCNC peut admettre des machines ayant jusqu'à 9 axes en définissant un ensemble supplémentaire de 3 axes linéaires, connus comme U, V et W. :lang: fr :toc:

## Chapitre 22

# Image-to-gcode: Usiner un depth maps



### 22.1 Qu'est-ce qu'un *depth map*?

Il s'agit d'une image en échelle de gris dont la luminosité de chaque pixel correspond à la profondeur (ou hauteur) de chaque point de l'objet.

### 22.2 Intégrer image-to-gcode dans l'interface utilisateur d'AXIS

Ajoutez les lignes suivantes dans la section: *[FILTER]* de votre fichier .ini pour qu'AXIS invoque automatiquement image-to-gcode à l'ouverture d'une image .png, .gif, ou .jpg:

```
PROGRAM_EXTENSION = .png,.gif,.jpg Grayscale Depth Image
```

Le fichier de configuration: *sim/axis.ini* est déjà configuré de cette façon.

## 22.3 Utilisation d'image-to-gcode

image-to-gcode peut être démarré soit en ouvrant une image dans AXIS, soit en invoquant image-to-gcode dans une console, de la manière suivante:

```
image-to-gcode torus.png > torus.ngc
```

Ajustez les réglages dans la colonne de droite, puis pressez *OK* pour créer le G-code. Selon la taille de l'image et les options choisies, le traitement peut durer de quelques secondes à quelques minutes. Quand une image est appelée, le G-code sera automatiquement chargé et visualisé dans AXIS une fois le traitement terminé. Dans AXIS, faites *Recharger* pour afficher de nouveau l'écran d'options d' image-to-gcode, vous pourrez ainsi travailler en boucle.

## 22.4 Les différentes options

### 22.4.1 Unités

Spécifie quelle unité sera utilisée dans le G-code généré G20 (pouces) ou G21 (mm), ce sera également l'unité utilisée par toutes les options marquées: (*units*).

### 22.4.2 Invert Image

Si *no*, le pixel noir sera le point le plus bas et le pixel blanc le point le plus haut. Si *yes*, le pixel noir sera le point le plus haut et le pixel blanc le point le plus bas.

### 22.4.3 Normalize Image

Si *yes*, le pixel le plus sombre est ramené au noir, le pixel le plus lumineux est ramené au blanc.

### 22.4.4 Expand Image Border

Si *None*, l'image entrée sera utilisée telle-quelle, les détails les plus aux bords de l'image pourraient être coupés. Si *White* ou *Black*, alors une bordure de pixels égale au diamètre de l'outil sera ajoutée sur tout le pourtour pour éviter ce risque.

### 22.4.5 Tolerance (unités)

Quand une série de points est proche d'une ligne droite au point d'être dans la *tolerance* , elle sera traitée comme une ligne droite en sortie. Augmenter la tolérance peut donner de meilleures performances de contourage avec LinuxCNC, mais peut aussi estomper ou gommer les détails les plus fins de l'image.

### 22.4.6 Pixel Size (unités)

Il y a beaucoup d'unités pour un pixel dans l'image entrée. Habituellement ce nombre est beaucoup plus petit que 1.0. Par exemple, pour usiner un objet de 50x50mm depuis une image de 400x400 pixels, utiliser un *pixel size* de 0.125, parce que  $50 / 400 = 0.125$ .

### 22.4.7 Plunge Feed Rate (unités par minute)

Vitesse du mouvement de plongée initial.

#### 22.4.8 Feed Rate (unités par minute)

Vitesse d'avance pour le reste de l'usinage.

#### 22.4.9 Spindle Speed (RPM)

Vitesse de rotation de la broche, en tours/mn

#### 22.4.10 Scan Pattern

Modèles de balayage possibles:

- Rangées
- Colonnes
- Rangées puis colonnes
- Colonnes puis rangées

#### 22.4.11 Scan Direction

Directions de balayage possibles:

- Positive: le fraisage commence à de petites valeurs de X ou Y et se poursuit avec des valeurs croissantes.
- Négative: le fraisage commence à des valeurs élevées de X ou Y et se poursuit avec des valeurs décroissantes.
- Alternative: le fraisage commence aux valeurs de X ou Y où s'est terminé le dernier mouvement. Cela réduit les déplacements *en l'air*.
- Up Milling: le fraisage commence en points bas et se poursuit vers les points hauts.
- Down Milling: le fraisage commence en points hauts et se poursuit vers les points bas.

#### 22.4.12 Depth (unités)

Le dessus du bloc est toujours à  $Z=0$ . La profondeur d'usinage dans le matériau est de  $Z=-depth$ .

#### 22.4.13 Step Over (pixels)

Distance entre rangées ou colonnes adjacentes. Pour trouver le nombre en pixels pour une distance donnée en unités, calculez: *distance/pixel size* et arrondissez au nombre le plus proche'. Par exemple: si *pixel size*=.006 et le pas souhaité sur la *distance*=.015, alors utilisez un Step Over de 2 ou 3 pixels, parce que  $.015/.006=2.5$ '.

#### 22.4.14 Tool Diameter

Le diamètre du taillant de l'outil.

#### 22.4.15 Safety Height

La hauteur à laquelle les mouvements de traversée. image-to-gcode considère toujours le dessus du matériau comme étant:  $Z=0$ .

#### 22.4.16 Tool Type

La forme du taillant de l'outil. Les formes possibles sont:

- Hémisphérique
- Plate
- Vé à 45 degrés
- Vé à 60 degrés

### 22.4.17 Lace bounding

Contrôle si les zones relativement plates le long d'une colonne ou d'une rangée peuvent être ignorées. Ces options n'ont de sens que pour un fraisage dans les deux directions. Trois choix sont possibles:

- None: toutes les rangées et les colonnes seront entièrement fraisées.
- Secondary: lors du fraisage dans la deuxième direction, les zones qui ne présentent pas une forte pente dans cette direction seront ignorées.
- Full: lors du fraisage dans la première direction, les zones qui présentent une forte pente dans la deuxième direction seront ignorées. Lors du fraisage dans la deuxième direction, les zones qui ne présentent pas une forte pente dans cette direction seront ignorées.

### 22.4.18 Contact angle

Quand *Lace bounding* n'est pas None, les pentes qui présentent une pente supérieure à *Contact angle* seront considérées comme de fortes pentes et celles en dessous de cet angle considérées comme de faibles pentes.

### 22.4.19 Offset d'ébauche et profondeur par passe d'ébauche

Image-to-gcode peut optionnellement produire des passes d'ébauche. La profondeur des passes d'ébauche successives est fixée par *Roughing depth per pass*. Par exemple, entrer 0.2 pour une première passe d'ébauche d'une profondeur de 0.2, la seconde passe d'ébauche aura une profondeur de 0.4 et ainsi de suite, jusqu'à ce que la profondeur totale Depth de l'image soit atteinte. Aucune des passes d'ébauche n'usinera plus près de la partie finale que Roughing Offset. La figure ci-dessous montre une grande profondeur verticale à usiner. Sur cette image, la profondeur des passes d'ébauche est de 0.2 pouces et Roughing Offset de 0.1 pouces.

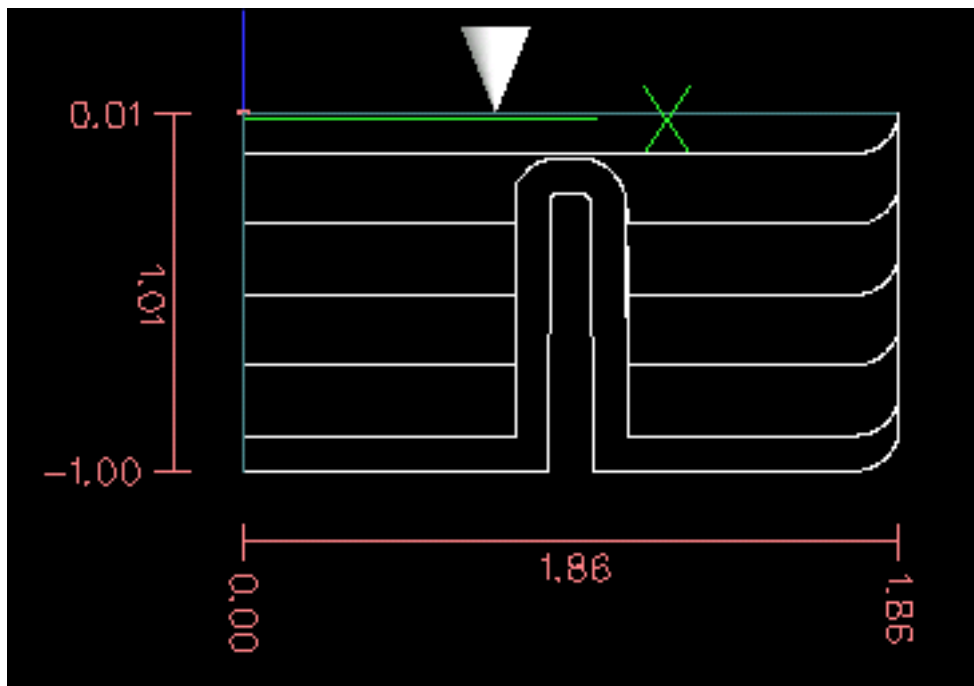


FIGURE 22.1 – Passes d'ébauche

## Chapitre 23

# Glossary

A listing of terms and what they mean. Some terms have a general meaning and several additional meanings for users, installers, and developers.

### **Acme Screw**

A type of lead-screw that uses an Acme thread form. Acme threads have somewhat lower friction and wear than simple triangular threads, but ball-screws are lower yet. Most manual machine tools use acme lead-screws.

### **Axis**

One of the computer controlled movable parts of the machine. For a typical vertical mill, the table is the X axis, the saddle is the Y axis, and the quill or knee is the Z axis. Angular axes like rotary tables are referred to as A, B, and C. Additional linear axes relative to the tool are called U, V, and W respectively.

### **Axis(GUI)**

One of the Graphical User Interfaces available to users of LinuxCNC. It features the modern use of menus and mouse buttons while automating and hiding some of the more traditional LinuxCNC controls. It is the only open-source interface that displays the entire tool path as soon as a file is opened.

### **Backlash**

The amount of "play" or lost motion that occurs when direction is reversed in a lead screw, or other mechanical motion driving system. It can result from nuts that are loose on leadscrews, slippage in belts, cable slack, "wind-up" in rotary couplings, and other places where the mechanical system is not "tight". Backlash will result in inaccurate motion, or in the case of motion caused by external forces (think cutting tool pulling on the work piece) the result can be broken cutting tools. This can happen because of the sudden increase in chip load on the cutter as the work piece is pulled across the backlash distance by the cutting tool.

### **Backlash Compensation**

Any technique that attempts to reduce the effect of backlash without actually removing it from the mechanical system. This is typically done in software in the controller. This can correct the final resting place of the part in motion but fails to solve problems related to direction changes while in motion (think circular interpolation) and motion that is caused when external forces (think cutting tool pulling on the work piece) are the source of the motion.

### **Ball Screw**

A type of lead-screw that uses small hardened steel balls between the nut and screw to reduce friction. Ball-screws have very low friction and backlash, but are usually quite expensive.

### **Ball Nut**

A special nut designed for use with a ball-screw. It contains an internal passage to re-circulate the balls from one end of the screw to the other.

### **CNC**

Computer Numerical Control. The general term used to refer to computer control of machinery. Instead of a human operator turning cranks to move a cutting tool, CNC uses a computer and motors to move the tool, based on a part program.

### **Comp**

A tool used to build, compile and install LinuxCNC HAL components.

---

### **Configuration(n)**

A directory containing a set of configuration files. Custom configurations are normally saved in the users home/LinuxCNC/configs directory. These files include LinuxCNC's traditional INI file and HAL files. A configuration may also contain several general files that describe tools, parameters, and NML connections.

### **Configuration(v)**

The task of setting up LinuxCNC so that it matches the hardware on a machine tool.

### **Coordinate Measuring Machine**

A Coordinate Measuring Machine is used to make many accurate measurements on parts. These machines can be used to create CAD data for parts where no drawings can be found, when a hand-made prototype needs to be digitized for moldmaking, or to check the accuracy of machined or molded parts.

### **Display units**

The linear and angular units used for onscreen display.

### **DRO**

A Digital Read Out is a system of position-measuring devices attached to the slides of a machine tool, which are connected to a numeric display showing the current location of the tool with respect to some reference position. DROs are very popular on hand-operated machine tools because they measure the true tool position without backlash, even if the machine has very loose Acme screws. Some DROs use linear quadrature encoders to pick up position information from the machine, and some use methods similar to a resolver which keeps rolling over.

### **EDM**

EDM is a method of removing metal in hard or difficult to machine or tough metals, or where rotating tools would not be able to produce the desired shape in a cost-effective manner. An excellent example is rectangular punch dies, where sharp internal corners are desired. Milling operations can not give sharp internal corners with finite diameter tools. A *wire* EDM machine can make internal corners with a radius only slightly larger than the wire's radius. A *sinker* EDM can make internal corners with a radius only slightly larger than the radius on the corner of the sinking electrode.

### **LinuxCNC**

The Enhanced Machine Controller. Initially a NIST project. LinuxCNC is able to run a wide range of motion devices.

### **LinuxCNCIO**

The module within LinuxCNC that handles general purpose I/O, unrelated to the actual motion of the axes.

### **LinuxCNCMOT**

The module within LinuxCNC that handles the actual motion of the cutting tool. It runs as a real-time program and directly controls the motors.

### **Encoder**

A device to measure position. Usually a mechanical-optical device, which outputs a quadrature signal. The signal can be counted by special hardware, or directly by the parport with LinuxCNC.

### **Feed**

Relatively slow, controlled motion of the tool used when making a cut.

### **Feed rate**

The speed at which a cutting motion occurs. In auto or mdi mode, feed rate is commanded using an F word. F10 would mean ten machine units per minute.

### **Feedback**

A method (e.g., quadrature encoder signals) by which LinuxCNC receives information about the position of motors

### **Feedrate Override**

A manual, operator controlled change in the rate at which the tool moves while cutting. Often used to allow the operator to adjust for tools that are a little dull, or anything else that requires the feed rate to be "tweaked".

### **Floating Point Number**

A number that has a decimal point. (12.300) In HAL it is known as float.

### **G-Code**

The generic term used to refer to the most common part programming language. There are several dialects of G-code, LinuxCNC uses RS274/NGC.

### **GUI**

Graphical User Interface.

### **General**

A type of interface that allows communications between a computer and a human (in most cases) via the manipulation of icons and other elements (widgets) on a computer screen.

### **LinuxCNC**

An application that presents a graphical screen to the machine operator allowing manipulation of the machine and the corresponding controlling program.

### **HAL**

Hardware Abstraction Layer. At the highest level, it is simply a way to allow a number of building blocks to be loaded and interconnected to assemble a complex system. Many of the building blocks are drivers for hardware devices. However, HAL can do more than just configure hardware drivers.

### **Home**

A specific location in the machine's work envelope that is used to make sure the computer and the actual machine both agree on the tool position.

### **ini file**

A text file that contains most of the information that configures LinuxCNC for a particular machine

### **Instance**

One can have an instance of a class or a particular object. The instance is the actual object created at runtime. In programmer jargon, the Lassi object is an instance of the Dog class.

### **Joint Coordinates**

These specify the angles between the individual joints of the machine. See also Kinematics

### **Jog**

Manually moving an axis of a machine. Jogging either moves the axis a fixed amount for each key-press, or moves the axis at a constant speed as long as you hold down the key. In manual mode, jog speed can be set from the graphical interface.

### **kernel-space**

See real-time.

### **Kinematics**

The position relationship between world coordinates and joint coordinates of a machine. There are two types of kinematics. Forward kinematics is used to calculate world coordinates from joint coordinates. Inverse kinematics is used for exactly the opposite purpose. Note that kinematics does not take into account, the forces, moments etc. on the machine. It is for positioning only.

### **Lead-screw**

An screw that is rotated by a motor to move a table or other part of a machine. Lead-screws are usually either ball-screws or acme screws, although conventional triangular threaded screws may be used where accuracy and long life are not as important as low cost.

### **Machine units**

The linear and angular units used for machine configuration. These units are specified and used in the ini file. HAL pins and parameters are also generally in machine units.

### **MDI**

Manual Data Input. This is a mode of operation where the controller executes single lines of G-code as they are typed by the operator.

### **NIST**

National Institute of Standards and Technology. An agency of the Department of Commerce in the United States.

### **Offsets**

An arbitrary amount, added to the value of something to make it equal to some desired value. For example, gcode programs are often written around some convenient point, such as X0, Y0. Fixture offsets can be used to shift the actual execution point of that gcode program to properly fit the true location of the vise and jaws. Tool offsets can be used to shift the "uncorrected" length of a tool to equal that tool's actual length.

### **Part Program**

A description of a part, in a language that the controller can understand. For LinuxCNC, that language is RS-274/NGC, commonly known as G-code.

### **Program Units**

The linear and angular units used in a part program. The linear program units do not have to be the same as the linear machine units. See G20 and G21 for more information. The angular program units are always measured in degrees.

### **Python**

General-purpose, very high-level programming language. Used in LinuxCNC for the Axis GUI, the Stepconf configuration tool, and several G-code programming scripts.

### **Rapid**

Fast, possibly less precise motion of the tool, commonly used to move between cuts. If the tool meets the workpiece or the fixturing during a rapid, it is probably a bad thing!

### **Rapid rate**

The speed at which a rapid motion occurs. In auto or mdi mode, rapid rate is usually the maximum speed of the machine. It is often desirable to limit the rapid rate when testing a g-code program for the first time.

### **Real-time**

Software that is intended to meet very strict timing deadlines. Under Linux, in order to meet these requirements it is necessary to install RTAI or RTLINUX and build the software to run in those special environments. For this reason real-time software runs in kernel-space.

### **RTAI**

Real Time Application Interface, see <https://www.rtai.org/>, one of two real-time extensions for Linux that LinuxCNC can use to achieve real-time performance.

### **RTLINUX**

See <http://www.rtlinux.org>, one of two real-time extensions for Linux that LinuxCNC can use to achieve real-time performance.

### **RTAPI**

A portable interface to real-time operating systems including RTAI and RTLINUX

### **RS-274/NGC**

The formal name for the language used by LinuxCNC part programs.

### **Servo Motor**

Generally, any motor that is used with error-sensing feedback to correct the position of an actuator. Also, a motor which is specially-designed to provide improved performance in such applications.

### **Servo Loop**

A control loop used to control position or velocity of an motor equipped with a feedback device.

### **Signed Integer**

A whole number that can have a positive or negative sign. In HAL it is known as s32. (A signed 32-bit integer has a usable range of -2,147,483,647 to +2,147,483,647.)

### **Spindle**

The part of a machine tool that spins to do the cutting. On a mill or drill, the spindle holds the cutting tool. On a lathe, the spindle holds the workpiece.

### **Spindle Speed Override**

A manual, operator controlled change in the rate at which the tool rotates while cutting. Often used to allow the operator to adjust for chatter caused by the cutter's teeth. Spindle Speed Override assumes that the LinuxCNC software has been configured to control spindle speed.

### **Stepconf**

An LinuxCNC configuration wizard. It is able to handle many step-and-direction motion command based machines. It writes a full configuration after the user answers a few questions about the computer and machine that LinuxCNC is to run on.

### **Stepper Motor**

A type of motor that turns in fixed steps. By counting steps, it is possible to determine how far the motor has turned. If the load exceeds the torque capability of the motor, it will skip one or more steps, causing position errors.

### **TASK**

The module within LinuxCNC that coordinates the overall execution and interprets the part program.

### **Tcl/Tk**

A scripting language and graphical widget toolkit with which several of LinuxCNCs GUIs and selection wizards were written.

### **Traverse Move**

A move in a straight line from the start point to the end point.

### **Units**

See "Machine Units", "Display Units", or "Program Units".

### **Unsigned Integer**

A whole number that has no sign. In HAL it is known as u32. (An unsigned 32-bit integer has a usable range of zero to 4,294,967,296.)

### **World Coordinates**

This is the absolute frame of reference. It gives coordinates in terms of a fixed reference frame that is attached to some point (generally the base) of the machine tool.

## Chapitre 24

# Legal Section

### 24.1 Copyright Terms

Copyright (c) 2000-2013 LinuxCNC.org

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and one Back-Cover Text: "This LinuxCNC Handbook is the product of several authors writing for linuxCNC.org. As you find it to be of value in your work, we invite you to contribute to its revision and growth." A copy of the license is included in the section entitled "GNU Free Documentation License". If you do not find the license you may order a copy from Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307

### 24.2 GNU Free Documentation License

GNU Free Documentation License Version 1.1, March 2000

Copyright © 2000 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

#### 0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other written document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

#### 1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you".

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that

could fall directly within that overall subject. (For example, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, whose contents can be viewed and edited directly and straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup has been designed to thwart or discourage subsequent modification by readers is not Transparent. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML designed for human modification. Opaque formats include PostScript, PDF, proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

## **2. VERBATIM COPYING**

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

## **3. COPYING IN QUANTITY**

If you publish printed copies of the Document numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a publicly-accessible computer-network location containing a complete Transparent copy of the Document, free of added material, which the general network-using public has access to download anonymously at no charge using public-standard network protocols. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

## **4. MODIFICATIONS**

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission. B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has less than five). C. State on the Title page the name of the publisher of the Modified Version, as the publisher. D. Preserve all the copyright notices of the Document. E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices. F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below. G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice. H. Include an unaltered copy of this License. I. Preserve the section entitled "History", and its title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence. J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission. K. In any section entitled "Acknowledgements" or "Dedications", preserve the section's title, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein. L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles. M. Delete any section entitled "Endorsements". Such a section may not be included in the Modified Version. N. Do not retitle any existing section as "Endorsements" or to conflict in title with any Invariant Section.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

## 5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections entitled "History" in the various original documents, forming one section entitled "History"; likewise combine any sections entitled "Acknowledgements", and any sections entitled "Dedications". You must delete all sections entitled "Endorsements."

## 6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

## 7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, does not as a whole count as a Modified Version of the Document, provided no compilation copyright is claimed for the compilation. Such a compilation is called an "aggregate", and this License does not apply to the other self-contained works thus compiled with the Document, on account of their being thus compiled, if they are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one quarter of the entire aggregate, the Document's Cover Texts may be placed on covers that surround only the Document within the aggregate. Otherwise they must appear on covers around the whole aggregate.

## 8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License provided that you also include the original English version of this License. In case of a disagreement between the translation and the original English version of this License, the original English version will prevail.

## 9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

## 10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.


Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

**ADDENDUM:** How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright (c) YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have no Invariant Sections, write "with no Invariant Sections" instead of saying which ones are invariant. If you have no Front-Cover Texts, write "no Front-Cover Texts" instead of "Front-Cover Texts being LIST"; likewise for Back-Cover Texts.

If your document contains nontrivial examples of program code, we recommend releasing these examples in  under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

## Chapitre 25

# Index

—  
.axisrc, [31](#)  
Éditeur externe, [31](#)  
Étendues du programme, [20](#)  
3 et 4, [177](#)  
7 et 8, [178](#)

### Numbers

2  
3 et 4, [177](#)  
6  
7 et 8, [178](#)

### A

A/U, [14](#), [49](#), [50](#)  
ABANDON, [11](#)  
acme screw, [195](#)  
Aperçu machine CNC, [78](#)  
Appel de fichier, [168](#)  
Arrêt d'urgence, [19](#)  
arrêt optionnel, [51](#), [79](#)  
Arrêts optionnels, [81](#)  
Arrosage, [24](#)  
arrosage, [52](#), [79](#), [80](#)  
Auto, [11](#), [50](#), [56](#), [59](#), [67](#)  
axes, [78](#)  
axes linéaires primaires, [78](#)  
axes linéaires secondaires, [78](#)  
Axes rotatifs, [78](#)  
AXIS, [13](#), [28](#)  
Axis, [13](#)  
axis, [195](#)  
AXIS avec un tour, [28](#)

### B

backlash, [195](#)  
backlash compensation, [195](#)  
ball nut, [195](#)  
ball screw, [195](#)  
Block Delete, [109](#)  
Boucles, [166](#)  
Bouton effacement de bloc, [79](#)  
break, [166](#)

Broche, [24](#)  
broche, [50](#), [52](#), [79](#)

### C

call, [165](#)  
Changement D'Outil Manuel, [28](#)  
chargement, [81](#)  
CNC, [3](#), [55](#), [195](#)  
Commentaires, [118](#)  
Commentaires spéciaux, [31](#)  
comp, [195](#)  
Compensation d'outil, [90](#)  
Concepts pour l'utilisateur, [71](#)  
Conditionnel: if  
    elseif  
    else, [167](#)  
continue, [166](#)  
Contrôle de trajectoire continue avec tolérance, [142](#)  
Contrôle manuel, [15](#), [22](#)  
coordinate measuring machine, [196](#)  
coordonnées polaires, [115](#)  
correcteur de vitesse, [11](#), [50](#)  
Correcteur de vitesse broche, [25](#)  
correcteur vitesse, [61](#)  
correcteur vitesse broche, [50](#), [79](#)  
Correcteurs de vitesse, [15](#), [25](#)  
correcteurs vitesse, [79](#)  
Cycles de perçage, [145](#)  
Cycles de perçage G81-G89, [145](#)

### D

display units, [196](#)  
do, [166](#)  
Données manuelles, [15](#)  
DRO, [196](#)

### E

EDM, [196](#)  
effacement de bloc, [81](#)  
else, [167](#)  
elseif, [167](#)  
    else, [167](#)  
encoder, [196](#)

endif, [167](#)  
endsub, [165](#)  
endwhile, [166](#)  
Enregistrement des mesures, [119](#)  
ESTOP, [11](#), [60](#)  
Expressions, [113](#)

## F

F: Réglage de la vitesse d'avance travail, [169](#)  
feed, [196](#)  
feed rate, [196](#)  
feedback, [196](#)  
feedrate override, [196](#)  
Fichier d'outils, [82](#)  
Fonctions, [113](#)  
Format de la table d'outils, [91](#)

## G

G-Code, [196](#)  
G-code, [108](#)  
G-Code bonnes pratiques, [121](#)  
G-codes modaux, [117](#)  
G0 Interpolation linéaire en vitesse rapide, [124](#)  
G1 Interpolation linéaire en vitesse travail, [125](#)  
G10 L1, [91](#)  
G10 L1 Ajustements dans la table d'outils, [132](#)  
G10 L10, [91](#)  
G10 L10 modifie les offsets d'outil dans la table d'outils, [133](#)  
G10 L11, [91](#)  
G10 L11 modifie les offsets d'outil dans la table d'outils, [134](#)  
G10 L2 Établissement de l'origine d'un système de coordonnées, [132](#)  
G10 L20 Établissement de l'origine d'un système de coordonnées, [134](#)  
G17 Plan XY, [135](#)  
G18 Plan XZ, [135](#)  
G19 Plan YZ, [135](#)  
G2 Interpolation circulaire sens horaire, [125](#)  
G20 Pouce, [135](#)  
G21 Millimètre, [135](#)  
G28, [135](#)  
G28.1, [135](#)  
G3 Interpolation circulaire anti-horaire, [125](#)  
G30, [135](#)  
G30.1, [135](#)  
G33 Mouvement avec broche synchronisée, [136](#)  
G33.1 Taraudage rigide, [137](#)  
G38.2 Palpeur, [138](#)  
G38.3 Palpeur, [138](#)  
G38.4 Palpeur, [138](#)  
G38.5 Palpeur, [138](#)  
G4 Temporisation, [130](#)  
G40 Révocation de la compensation de rayon, [139](#)  
G41 Compensation d'outil, [139](#)  
G41.1 Compensation dynamique, [139](#)

G42 Compensation d'outil, [139](#)  
G42.1 Compensation dynamique, [139](#)  
G43 Activation de la compensation de longueur d'outil, [140](#)  
G43.1 Compensation dynamique de longueur d'outil, [140](#)  
G49 Révocation de compensation de longueur d'outil, [140](#)  
G5.1 B-spline quadratique, [130](#)  
G5.2 G5.3 NURBs Block, [130](#)  
G53 Mouvement en coordonnées absolues, [141](#)  
G55, [85](#)  
G61 Trajectoire exacte, [142](#)  
G61.1 Arrêt exact, [142](#)  
G7 Mode diamètre sur les tours, [132](#)  
G73 Cycle de perçage avec brise copeaux, [142](#)  
G76 Cycle de filetage multi-passe, [143](#)  
G8 Mode rayon sur les tours, [132](#)  
G80 Révocation des codes modaux, [147](#)  
G81 Cycle de perçage, [148](#)  
G81-G89  
    Cycles de perçage, [145](#)  
G82 Cycle de perçage avec tempo, [151](#)  
G83 Cycle de perçage avec déburrage, [151](#)  
G84 Cycle de taraudage, [152](#)  
G85 Cycle d'alésage, [152](#)  
G86 Cycle d'alésage, [152](#)  
G87 Alésage inverse, [152](#)  
G88 Cycle d'alésage, [152](#)  
G89 Cycle d'alésage avec tempo, [153](#)  
G90 Mode de déplacement absolu, [154](#)  
G91 Mode de déplacement relatif, [154](#)  
G92 Décalages d'origine des systèmes de coordonnées, [155](#)  
G93  
    G94  
        G95: Choix des modes de vitesse, [155](#)  
G94  
    G95: Choix des modes de vitesse, [155](#)  
G95: Choix des modes de vitesse, [155](#)  
G96  
    G97: Vitesse de coupe constante  
        Vitesse de coupe en tr/mn, [156](#)  
G97: Vitesse de coupe constante  
    Vitesse de coupe en tr/mn, [156](#)  
G98  
    G99 Retrait à la position initiale  
        Retrait sur R, [156](#)  
G99 Retrait à la position initiale  
    Retrait sur R, [156](#)  
Gouttelettes, [50](#)  
Groupes modaux, [117](#)  
GUI, [195](#), [196](#)

## H

HAL, [197](#)  
home, [197](#)

## I

if, [167](#)  
Indirection, [168](#)

INI, [197](#)  
Instance, [197](#)  
Interraction vitesse, [81](#)

## J

jog, [197](#)  
joint coordinates, [197](#)

## K

KEYSTICK, [68](#)  
kinematics, [197](#)

## L

lancer, [51](#)  
Le G-code, [123](#)  
lead screw, [197](#)  
Les autres codes, [169](#)  
Les M-codes, [158](#)  
Les nombres, [110](#)  
Linux, [4](#)  
LinuxCNC, [196](#)  
LinuxCNC User Introduction, [3](#)  
LinuxCNCIO, [196](#)  
LinuxCNCMOT, [196](#)  
Log général, [119](#)  
loop, [198](#)

## M

M-codes définis par l'utilisateur M100-M199, [163](#)  
M-codes modaux, [118](#)  
M0 Pause dans le programme, [158](#)  
M1 Pause optionnelle dans le programme, [158](#)  
M100 à M199 M-codes définis par l'utilisateur, [163](#)  
M2 Fin de programme, [159](#)  
M3 Broche en sens horaire, [159](#)  
M30 Fin de programme avec déchargement pièce, [159](#)  
M4 Broche en sens anti-horaire, [159](#)  
M48  
    M49 Autoriser/Inhiber les correcteurs de vitesse, [160](#)  
M49 Autoriser/Inhiber les correcteurs de vitesse, [160](#)  
M5 Arrêt de broche, [159](#)  
M50 Contrôle du correcteur de vitesse travail, [160](#)  
M51 Contrôle du correcteur de vitesse broche, [161](#)  
M52 Contrôle vitesse adaptative, [161](#)  
M53 Contrôle coupure vitesse, [161](#)  
M6 Appel d'outil, [159](#)  
M60 Pause pour déchargement pièce, [159](#)  
M61 Correction du numéro de l'outil courant, [161](#)  
M62 Contrôle un bit de sortie numérique, [161](#)  
M66 Contrôle d'entrée numérique et analogique, [162](#)  
M67 Contrôle de sortie analogique synchronisée avec un mouvement, [162](#)  
M68 Contrôle de Sortie analogique directe, [163](#)  
M7 Arrosage gouttelettes, [160](#)  
M8 Arrosage fluide, [160](#)  
M9 Arrêt des arrosages, [160](#)  
machine units, [197](#)  
Manuel, [11](#), [50](#), [58](#), [66](#)

Marche/Arrêt, [19](#)  
MDI, [11](#), [24](#), [50](#), [60](#), [197](#)  
MDI), [50](#)  
Messages, [119](#)  
Messages de débogage, [119](#)  
Mini, [55](#)  
mini, [55](#)  
mots, [109](#)

## N

NGCGUI, [33](#)  
NIST, [197](#)  
Numéro de ligne, [109](#)

## O

O-codes, [165](#)  
offsets, [197](#)  
Opérateurs binaires, [113](#)  
Opérations unaires, [113](#)  
OpenGL, [13](#)  
Ordre d'exécution, [120](#)  
Orientations des outils de tour, [176](#)  
Origine Machine, [49](#)  
Origine Piece, [50](#)  
Outils en positions 1  
    2  
        3 et 4, [177](#)  
Outils en positions 5  
    6  
        7 et 8, [178](#)  
ouvrir, [51](#)

## P

Panneau de contrôle virtuel, [31](#)  
Paramètres, [110](#)  
paramètres, [82](#)  
Paramètres numérotés, [111](#)  
Paramètres système, [113](#)  
Parcours d'outil, [20](#)  
parcours outil, [56](#)  
part Program, [197](#)  
Particularités des tours, [176](#)  
pas a pas, [51](#)  
pause, [51](#)  
point contrôlé, [79](#)  
Position: Absolue, [15](#)  
Position: Actuelle, [15](#)  
Position: Commandée, [15](#)  
Position: Relative, [15](#)  
Précédence des opérateurs, [113](#)  
program units, [197](#)  
Programmation RS274/NGC, [188](#)  
Python, [13](#), [28](#)

## R

rapid, [198](#)  
rapid rate, [198](#)

rapide, [124](#)  
real-time, [198](#)  
Repeat, [167](#)  
reprise, [51](#)  
Retrait sur R, [156](#)  
return, [165](#)  
RS274NGC, [198](#)  
RTAI, [198](#)  
RTAPI, [198](#)  
RTLINUX, [198](#)

## S

S: Réglage de la vitesse de rotation de la broche, [169](#)  
servo motor, [198](#)  
Sherline, [55](#)  
Signed Integer, [198](#)  
Sous-programmes, [165](#)  
spindle, [198](#)  
stepper motor, [198](#)  
sub, [165](#)  
Systèmes de coordonnées, [84](#)

## T

T: Choix de l'outil, [169](#)  
Table des index du G Code, [123](#)  
TASK, [198](#)  
Tcl, [48](#)  
tempo, [80](#)  
Tk, [13](#), [48](#), [198](#)  
TkLinuxCNC, [48](#)  
tklinuxcnc, [48](#)  
Toucher, [15](#), [90](#)  
Touchy GUI, [45](#)  
trace d'outil, [63](#)  
Trajectoire contrôlée, [142](#)  
Traverse Move, [198](#)

## U

unités, [80](#)  
units, [199](#)  
Unsigned Integer, [199](#)

## V

vérifier, [51](#)  
vitesse d'avance, [80](#)  
Vitesse de coupe en tr/mn, [156](#)  
Vitesse de jog, [25](#)  
vitesse de jog, [50](#)  
Vitesse maxi, [25](#)  
Vue générale du G-code de LinuxCNC, [108](#)

## W

while, [166](#)  
world coordinates, [199](#)

---