

# **AmiCAD**

FLORAC Roland

<b>COLLABORATORS</b>
----------------------

	<i>TITLE :</i> AmiCAD		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	FLORAC Roland	July 31, 2024	

<b>REVISION HISTORY</b>
-------------------------

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>AmiCAD</b>	<b>1</b>
1.1	Sommaire . . . . .	1
1.2	Distribution . . . . .	1
1.3	Présentation, exigences . . . . .	2
1.4	Installation . . . . .	2
1.5	Lancement du programme / ToolTypes . . . . .	3
1.6	Les divers objets manipulés par AmiCAD . . . . .	4
1.7	Les lignes . . . . .	4
1.8	Les connexions . . . . .	5
1.9	Composants . . . . .	5
1.10	Les Références affectées aux composants . . . . .	6
1.11	Les valeurs et les types affectés aux composants . . . . .	6
1.12	Les labels . . . . .	6
1.13	Les connecteurs . . . . .	7
1.14	Les textes . . . . .	7
1.15	Les boîtes (rectangles) . . . . .	8
1.16	Les ellipses . . . . .	8
1.17	Les triangles . . . . .	9
1.18	Arcs de cercles . . . . .	9
1.19	Les groupes . . . . .	10
1.20	Barre de titre . . . . .	10
1.21	Type d'outil HELPFILE . . . . .	10
1.22	Type d'outil WINDOW . . . . .	10
1.23	Type d'outil STARTUP . . . . .	11
1.24	Type d'outil LIBS . . . . .	11
1.25	Type d'outil CLIPS . . . . .	11
1.26	Type d'outil MACRO . . . . .	11
1.27	Type d'outil X_ICON . . . . .	11
1.28	Type d'outil Y_ICON . . . . .	12
1.29	Type d'outil SHEET_WIDTH . . . . .	12

---

1.30	Type d'outil SHEET_HEIGHT . . . . .	12
1.31	Type d'outil GRIDSIZE . . . . .	12
1.32	Les menus AmiCAD . . . . .	12
1.33	Menu Projet . . . . .	13
1.34	Menu Projet/Charger . . . . .	13
1.35	Menu Projet/Sauver . . . . .	14
1.36	Menu Projet/Sauver en . . . . .	14
1.37	Menu Projet/Sauver format IFF . . . . .	14
1.38	Menu Projet/Renommer . . . . .	14
1.39	Menu Projet/Annoter fichier . . . . .	15
1.40	Menu Projet/Supprimer fichier . . . . .	15
1.41	Menu Projet/Réduire fenêtre . . . . .	15
1.42	Menu Projet/Cacher fenêtre . . . . .	16
1.43	Menu Projet/Autre fenêtre . . . . .	16
1.44	Menu Projet/Initialiser . . . . .	16
1.45	Menu Projet/Imprimer . . . . .	17
1.46	Menu Projet/Informations . . . . .	17
1.47	Menu Projet/Schémas . . . . .	17
1.48	Menu Projet/Aide . . . . .	18
1.49	Menu Projet/Quitter . . . . .	18
1.50	Icône d'application AmiCAD . . . . .	18
1.51	Gestion de la mémoire . . . . .	19
1.52	Menu Dessin . . . . .	19
1.53	Menu Dessin/Pas de la grille= 10 . . . . .	20
1.54	Menu Dessin/Aligner sur grille . . . . .	20
1.55	Menu Dessin/Choisir composant . . . . .	20
1.56	Menu Dessin/Placer composant . . . . .	21
1.57	Menu Dessin/Placer référence . . . . .	21
1.58	Menu Dessin/Placer valeur/type . . . . .	22
1.59	Menu Dessin/Placer numéros bornes . . . . .	22
1.60	Menu Dessin/Rotation . . . . .	22
1.61	Menu Dessin/Symétrie . . . . .	23
1.62	Menu Dessin/Alternier symbole . . . . .	23
1.63	Menu Dessin/Position normale . . . . .	24
1.64	Menu Dessin/Placer fil . . . . .	24
1.65	Menu Dessin/Tracé orthogonal . . . . .	24
1.66	Menu Dessin/Tracé continu . . . . .	25
1.67	Menu Dessin/Trait double . . . . .	25
1.68	Menu Dessin/Trait fort (Bus) . . . . .	25

---

1.69 Menu Dessin/Largeur quelconque . . . . .	26
1.70 Menu Dessin/Pointillés . . . . .	26
1.71 Menu Dessin/Placer rectangle . . . . .	26
1.72 Menu Dessin/Placer triangle . . . . .	27
1.73 Menu Dessin/Placer ellipse . . . . .	27
1.74 Menu Dessin/Surfaces pleines . . . . .	28
1.75 Menu Dessin/Placer arc de cercle . . . . .	28
1.76 Menu Dessin/Placer connexion . . . . .	29
1.77 Menu Dessin/Placer texte . . . . .	29
1.78 Menu Dessin/Placer connecteur entrée . . . . .	29
1.79 Menu Dessin/Placer connecteur sortie . . . . .	30
1.80 Menu Dessin/Redessiner tout . . . . .	31
1.81 Les clips AmiCAD . . . . .	31
1.82 Menu Édition . . . . .	31
1.83 Menu Édition/Copier . . . . .	32
1.84 Menu Édition/Coller . . . . .	32
1.85 Menu Édition/Couper . . . . .	33
1.86 Menu Édition/Effacer . . . . .	33
1.87 Menu Édition/Cloner . . . . .	33
1.88 Menu Édition/Fixer sur grille . . . . .	33
1.89 Menu Édition/Passer devant . . . . .	34
1.90 Menu Édition/Passer derrière . . . . .	34
1.91 Menu Édition/Doubler taille . . . . .	34
1.92 Menu Édition/Diviser taille . . . . .	35
1.93 Menu Édition/Étirer . . . . .	35
1.94 Menu Édition/Grouper . . . . .	35
1.95 Menu Édition/Séparer . . . . .	36
1.96 Menu Édition/Sauver clip . . . . .	36
1.97 Menu Édition/Charger clip . . . . .	36
1.98 Menu Édition/Multisélection . . . . .	37
1.99 Menu Édition/Restaurer . . . . .	37
1.100 Menu Scripts . . . . .	37
1.101 Menu Macros . . . . .	37
1.102 Menu Macros/Ajouter menu . . . . .	38
1.103 Menu Macros/Éditer label . . . . .	38
1.104 Menu Utilisateur/Éditer macro . . . . .	38
1.105 Menu Utilisateur/Supprimer menu . . . . .	38
1.106 Menu Macros/Mode direct . . . . .	39
1.107 Menu Macros/Appel script . . . . .	39

1.108Menu Macros/ARexx...	40
1.109Port ARexx	40
1.110Menu Préférences	40
1.111Menu Préférences/Échelle objets	41
1.112Menu Préférences/Défilement auto	41
1.113Menu Préférences/Copie schéma sauvé	41
1.114Menu Préférences/Sauver icône	42
1.115Menu Préférences/Tirer lignes	42
1.116Menu Préférences/Afficher grille	42
1.117Menu Préférences/Nom complet	42
1.118Menu Préférences/Affichage	43
1.119Menu Préférences/Échelle horizontale	43
1.120Menu Préférences/Échelle verticale	43
1.121Menu Préférences/Dimensions document	43
1.122Menu Préférences/Palette	44
1.123Menu Préférences/Touches de fonction	44
1.124Touches - Raccourcis clavier	44
1.125Menu Préférences/Affichage/0- Largeur écran	45
1.126Menu Préférences/Affichage/4- Hauteur écran	45
1.127Menu Préférences/Affichage/0- Échelle 1	45
1.128Menu Préférences/Affichage/2- Zoom boîte	45
1.129Menu Préférences/Affichage/5- Recentrer	46
1.130Menu Préférences/Affichage/6- Doubler échelle	46
1.131Menu Préférences/Affichage/9- Diviser échelle	46
1.132Menu Préférences/Affichage/3- Saisie échelle	46
1.133Format des fichiers de configuration	46
1.134Écriture d'une ligne de commande(s)	47
1.135Les variables	48
1.136Les variables numériques	49
1.137Les chaînes de caractères	49
1.138Structure des scripts ARexx	49
1.139Liste des fonctions ARexx	50
1.140Classement thématique des fonctions ARexx	53
1.141Fonctions ARexx de traitement des chaînes de caractères	54
1.142Fonctions ARexx permettant le placement de nouveaux objets	54
1.143Fonctions ARexx permettant de gérer les blocs d'objets	54
1.144Fonctions ARexx permettant de gérer les objets	54
1.145Fonctions ARexx de gestion des préférences	56
1.146Fonctions ARexx de calcul	56

1.147Fonctions ARexx permettant le dialogue avec l'utilisateur . . . . .	56
1.148Fonctions ARexx diverses . . . . .	57
1.149Fonctions ARexx de gestion des fenêtres . . . . .	57
1.150Fonction ARexx DEF . . . . .	58
1.151Fonction ARexx ABS . . . . .	58
1.152Fonction ARexx ARC . . . . .	59
1.153Fonction ARexx ARRAYDIM . . . . .	59
1.154Fonction ARexx ASC . . . . .	60
1.155Fonction obsolète . . . . .	60
1.156Fonction ARexx ASKNUM . . . . .	60
1.157Fonction ARexx ASKTEXT . . . . .	61
1.158Fonction ARexx BLINK . . . . .	61
1.159Fonction ARexx BOX . . . . .	61
1.160Fonction ARexx CALL . . . . .	61
1.161Fonction ARexx CHR . . . . .	62
1.162Fonction ARexx CLIPPATH . . . . .	62
1.163Fonction ARexx CLIPUNIT . . . . .	63
1.164Fonction ARexx CLOSE . . . . .	63
1.165Fonction ARexx COL . . . . .	64
1.166Fonction ARexx CONVERT . . . . .	64
1.167Fonction ARexx COORDS . . . . .	64
1.168Fonction ARexx COPY . . . . .	65
1.169Fonction ARexx CURSMODE . . . . .	65
1.170Fonction ARexx DATE . . . . .	65
1.171Fonction ARexx DELARRAY . . . . .	66
1.172Fonction ARexx DELETE . . . . .	66
1.173Fonction ARexx DEVPINS . . . . .	66
1.174Fonction ARexx DIMARRAY . . . . .	67
1.175Fonction ARexx DIMSHEET . . . . .	67
1.176Fonction ARexx DRAW . . . . .	67
1.177Fonction ARexx DRAWMODE . . . . .	68
1.178Fonction ARexx EDIT . . . . .	68
1.179Fonction ARexx ELLIPSE . . . . .	68
1.180Fonction ARexx ENDCOL . . . . .	69
1.181Fonction ARexx ENDLINE . . . . .	69
1.182Fonction ARexx EXEC . . . . .	69
1.183Fonction ARexx FILENAME . . . . .	70
1.184Fonction ARexx FILEPART . . . . .	70
1.185Fonction ARexx FINDLAB . . . . .	70

1.186Fonction ARexx FINDLINE . . . . .	71
1.187Fonction ARexx FINDOBJ . . . . .	71
1.188Fonction ARexx FINDPART . . . . .	72
1.189Fonction ARexx FINDREF . . . . .	73
1.190Fonction ARexx FINDVAL . . . . .	73
1.191Fonction ARexx FIRSTSEL . . . . .	74
1.192Fonction ARexx FONTNAME . . . . .	74
1.193Fonction ARexx FONTSIZE . . . . .	74
1.194Fonction ARexx FOR . . . . .	74
1.195Fonction ARexx GETARRAY . . . . .	75
1.196Fonction ARexx GETCOLOR . . . . .	75
1.197Fonction ARexx GETDEVS . . . . .	76
1.198Fonction ARexx GETLABEL . . . . .	76
1.199Fonction ARexx GETLINE . . . . .	76
1.200Fonction ARexx GETNET . . . . .	77
1.201Fonction obsolète . . . . .	77
1.202Fonction ARexx GETPOINT . . . . .	77
1.203Fonction ARexx GETPOS . . . . .	78
1.204Fonction ARexx GETREF . . . . .	78
1.205Fonction ARexx GETVAL . . . . .	78
1.206Fonction ARexx GETZONE . . . . .	79
1.207Fonction ARexx GROUP . . . . .	79
1.208Fonction ARexx HEIGHT . . . . .	79
1.209Fonction ARexx HELP . . . . .	80
1.210Fonction ARexx HSCALE . . . . .	80
1.211Fonction ARexx IF . . . . .	80
1.212Fonction ARexx INIT . . . . .	80
1.213Fonction ARexx INPUT . . . . .	81
1.214Fonction ARexx JUNCTION . . . . .	81
1.215Fonction ARexx LANGUAGE . . . . .	82
1.216Fonction ARexx LEN . . . . .	82
1.217Fonction ARexx LIBRARY . . . . .	82
1.218Fonction ARexx LIBSPATH . . . . .	82
1.219Fonction ARexx LINE . . . . .	83
1.220Fonction ARexx LINKLAB . . . . .	83
1.221Fonction ARexx LINKREF . . . . .	83
1.222Fonction ARexx LINKVAL . . . . .	84
1.223Fonction ARexx LOAD . . . . .	84
1.224Fonction ARexx LOADCLIP . . . . .	84

---



1.225Fonction ARexx LOADKEYS . . . . .	84
1.226Fonction ARexx LOADLIB . . . . .	85
1.227Fonction ARexx LOADPREF . . . . .	85
1.228Fonction ARexx LOCK . . . . .	85
1.229Fonction ARexx MACRO . . . . .	85
1.230Fonction ARexx MAP . . . . .	86
1.231Fonction ARexx MARK . . . . .	86
1.232Fonction ARexx MARKZONE . . . . .	87
1.233Fonction ARexx MENU . . . . .	87
1.234Fonction ARexx MESSAGE . . . . .	87
1.235Fonction ARexx MESURE . . . . .	88
1.236Fonction ARexx MODIF . . . . .	88
1.237Fonction ARexx MOVE . . . . .	88
1.238Fonction ARexx NBSHEET . . . . .	89
1.239Fonction ARexx NBSHEETS . . . . .	89
1.240Fonction ARexx NEW . . . . .	89
1.241Fonction ARexx NEWSHEET . . . . .	89
1.242Fonction ARexx NEXTSEL . . . . .	90
1.243Fonction ARexx OBJECTS . . . . .	90
1.244Fonction ARexx OPEN . . . . .	90
1.245Fonction ARexx OUTPUT . . . . .	91
1.246Fonction ARexx PARTNAME . . . . .	91
1.247Fonction ARexx PASTE . . . . .	92
1.248Fonction ARexx PENWIDTH . . . . .	92
1.249Fonction ARexx PICKOBJ . . . . .	92
1.250Fonction ARexx PINCOL . . . . .	93
1.251Fonction ARexx PINLINE . . . . .	93
1.252Fonction ARexx PINNUM . . . . .	93
1.253Fonction ARexx PRINT . . . . .	93
1.254Fonction ARexx PUTPART . . . . .	94
1.255Fonction ARexx READCONV . . . . .	94
1.256Fonction ARexx READDEF . . . . .	94
1.257Fonction ARexx READDEV . . . . .	95
1.258Fonction ARexx READMAP . . . . .	95
1.259Fonction ARexx READTEXT . . . . .	96
1.260Fonction ARexx REMLIB . . . . .	96
1.261Fonction ARexx REQFILE . . . . .	96
1.262Fonction ARexx REQLIST . . . . .	97
1.263Fonction ARexx REQSHEET . . . . .	97

---

1.264Fonction ARexx REQUEST . . . . .	97
1.265Fonction ARexx RESET . . . . .	97
1.266Fonction ARexx REXXPORT . . . . .	98
1.267Fonction ARexx ROTATE . . . . .	98
1.268Fonction ARexx SAVE . . . . .	98
1.269Fonction ARexx SAVEALL . . . . .	99
1.270Fonction ARexx SAVECLIP . . . . .	99
1.271Fonction ARexx SAVECOPY . . . . .	99
1.272Fonction ARexx SAVEICON . . . . .	99
1.273Fonction ARexx SAVEIFF . . . . .	100
1.274Fonction ARexx SAVEKEYS . . . . .	100
1.275Fonction ARexx SAVEPREF . . . . .	100
1.276Fonction ARexx SCREEN . . . . .	100
1.277Fonction ARexx SCRMODE . . . . .	101
1.278Fonction ARexx SECURITY . . . . .	101
1.279Fonction ARexx SELECT . . . . .	101
1.280Fonction ARexx SELFILE . . . . .	102
1.281Fonction ARexx SELSHEET . . . . .	102
1.282Fonction ARexx SETARRAY . . . . .	102
1.283Fonction ARexx SETCOLOR . . . . .	103
1.284Fonction ARexx SETDEV . . . . .	103
1.285Fonction ARexx SETFILL . . . . .	103
1.286Fonction ARexx SETGRID . . . . .	104
1.287Fonction ARexx SETLABEL . . . . .	104
1.288Fonction ARexx SETORTHO . . . . .	104
1.289Fonction ARexx SETPINS . . . . .	105
1.290Fonction ARexx SETREF . . . . .	105
1.291Fonction ARexx SETSCALE . . . . .	106
1.292Fonction ARexx SETTEXT . . . . .	106
1.293Fonction ARexx SETVAL . . . . .	106
1.294Fonction ARexx SGN . . . . .	107
1.295Fonction ARexx SHEIGHT . . . . .	107
1.296Fonction ARexx SNAPGRID . . . . .	107
1.297Fonction ARexx STOBACK . . . . .	108
1.298Fonction ARexx STOFront . . . . .	108
1.299Fonction ARexx STR . . . . .	108
1.300Fonction ARexx SWIDTH . . . . .	108
1.301Fonction ARexx SYMMETRY . . . . .	108
1.302Fonction ARexx TEST . . . . .	109

1.303Fonction ARexx TIME . . . . .	109
1.304Fonction ARexx TITLE . . . . .	109
1.305Fonction ARexx TOOLBAR . . . . .	110
1.306Fonction ARexx TRIANGLE . . . . .	110
1.307Fonction ARexx TXHEIGHT . . . . .	111
1.308Fonction ARexx TXWIDTH . . . . .	111
1.309Fonction ARexx TYPE . . . . .	111
1.310Fonction ARexx UNGROUP . . . . .	112
1.311Fonction ARexx UNLINK . . . . .	112
1.312Fonction ARexx UNLOCK . . . . .	113
1.313Fonction ARexx UNMAP . . . . .	113
1.314Fonction ARexx UNMARK . . . . .	113
1.315Fonction ARexx VAL . . . . .	114
1.316Fonction ARexx VERSION . . . . .	114
1.317Fonction ARexx VSCALE . . . . .	114
1.318Fonction ARexx WHEIGHT . . . . .	115
1.319Fonction ARexx WHILE . . . . .	115
1.320Fonction ARexx WIDTH . . . . .	115
1.321Fonction ARexx WINDOW . . . . .	115
1.322Fonction ARexx WRITE . . . . .	116
1.323Fonction ARexx WTOBACK . . . . .	116
1.324Fonction ARexx WTOFRONT . . . . .	116
1.325Fonction ARexx WWIDTH . . . . .	117
1.326Fonction ARexx ZOOM . . . . .	117
1.327Commandes clavier . . . . .	117
1.328Utilisation des touches de fonction . . . . .	118
1.329gtlayout.library . . . . .	118
1.330Aide en ligne . . . . .	118
1.331Quelques macros utiles . . . . .	119
1.332BUG(s) ? (mais oui ! sûrement... (malheureusement !)) . . . . .	120
1.333Historique . . . . .	120
1.334Aidez-moi! . . . . .	124
1.335AMÉLIORATIONS POSSIBLES . . . . .	124
1.336Traduction . . . . .	124
1.337AmiCAD2META . . . . .	125
1.338L'auteur . . . . .	127
1.339Index . . . . .	127

# Chapter 1

## AmiCAD

### 1.1 Sommaire

AmiCAD Version 2.08 31 mars 2001

DISTRIBUTION	Copyright...
PRÉSENTATION, EXIGENCES	En grandes lignes...
INSTALLATION	Rapide...
LANCEMENT	Comment démarrer (Shell ou Workbench)
La barre de titre	Ce qu'on y trouve
Les différents objets	
Fichiers de configuration	Format des fichiers AmiCAD.prefs et AmiCAD.keys
LES MENUS	
Commandes~clavier	
Commandes ARexx	
Liste alphabétique	
Liste thématique	
Les scripts	
Les nombres	Comment ils sont traités.
Chaînes de caractères	Comment elles doivent être spécifiées.
BUGS (?)	
AMÉLIORATIONS futures	Rêvons un peu.
Historique	Quelle histoire...
L'auteur	
Help me!	
Quelques macros utiles	Toujours pratiques.
Traduction	
AmiCAD2META	Appel aux courageux...
	Un petit extra.

### 1.2 Distribution

Le programme AmiCAD a été conçu et écrit par R.Florac. Il est

donc sous Copyright R.Florac, il n'appartient pas au domaine public.

Ce programme est giftware/emailware. Son utilisation personnelle est libre. Cependant son utilisation commerciale ou professionnelle est interdite sans mon accord. Envoyez-moi vos suggestions quand à l'amélioration du programme ainsi que les éventuels rapports de bugs.

Vous pouvez me contacter par courrier électronique.

La distribution de ce programme est interdite sans autorisation, Aminet, seul, peut le distribuer sur son réseau et ses CD.

## 1.3 Présentation, exigences

AmiCAD est un programme écrit en langage C (SAS C Compiler 6.58), avec aussi un peu d'assembleur (Devpac 3.14).

Le programme ne fonctionne que sous système 3.0 au moins, avec un processeur 68020 ou mieux.

C'est un éditeur de schémas vectoriel. Cependant il ne travaille qu'avec des nombres entiers, pour plus de rapidité (et surtout une plus grande simplicité...). Il possède une interface ARexx, avec plus de 150 commandes, avec la possibilité de définir ses propres fonctions et d'utiliser des variables. D'autre part certaines fonctions font appel à la bibliothèque gtlayout.library version 42 ou plus. Celle-ci n'est pas distribuée avec le logiciel, vous pouvez vous la procurer dans le domaine public (site ou CR-ROM Aminet par exemple). Cependant le programme peut parfaitement fonctionner sans elle (seule la sélection des composants dans les bibliothèques sera un peu plus difficile).

Ce programme permet de travailler sur un nombre de fenêtres théoriquement illimité (sauf par la mémoire disponible !). Vous pouvez bénéficier des fonctions avancées du système 3.0: AppIcon, Pools...

## 1.4 Installation

Le programme peut être copié dans tout répertoire, à votre convenance, sous réserve qu'il y ait assez de place libre. Les fichiers de configuration AmiCAD.prefs et AmiCAD.keys doivent être copiés dans le même répertoire que le programme AmiCAD. Les fichiers contenant les symboles utilisés pour dessiner les schémas peuvent être copiés dans le même répertoire que le programme AmiCAD.

Le fichier d'aide AmiCAD.guide peut être copié dans tout emplacement à votre convenance, vous devrez cependant situer ce fichier à l'aide du tooltype HELPFIELD de l'icône du programme si vous voulez bénéficier de l'aide en ligne.

ATTENTION: le programme, pour fonctionner, fait appel aux ROM 3.0, son fonctionnement est impossible avec une version du système

---

antérieure.

## 1.5 Lancement du programme / ToolTypes

La taille de la pile courante du CLI (ou du Shell) peut être fixée à 4 ko (suffisant).

AmiCAD peut être lancé depuis le CLI ou le Workbench (en cliquant deux fois sur son icône).

Lancement depuis le CLI

Le modèle est le suivant:

AmiCAD FILE/M,HELPPFILE/K,LIBS/K,CLIPS/K,STARTUP/K

Les jokers AmigaDOS sont acceptés :

AmiCAD #?.schéma ==> tous les fichiers comportant l'extension ".schéma" seront chargés chacun dans une fenêtre (à condition qu'il y ait suffisamment de mémoire bien sûr).

Vous pouvez bien sûr lancer le programme sans spécifier de nom de fichier (le nom "Innomé" sera alors utilisé par défaut).

Le mot-clé HELPPFILE permet de spécifier où est situé le fichier d'aide:

Exemple: AmiCAD HELPPFILE HELP:AmiCAD.guide

Le mot-clé LIBS permet de spécifier où sont situées les bibliothèques de symboles:

Exemple: AmiCAD LIBS Work:AmiCAD/Symboles

Le mot-clé CLIPS permet de spécifier où sont situés les clips:

Exemple: AmiCAD CLIPS Work:AmiCAD/Clips

Le mot-clé STARTUP permet de spécifier un script ARexx à exécuter aussitôt après le lancement du programme.

Exemple: AmiCAD STARTUP démarrage.AmiCAD

Bien sûr tous ces mots-clé peuvent être combinés lors du même appel du programme:

AmiCAD HELPPFILE HELP:AmiCAD.guide LIBS Work:AmiCAD/Symboles Schéma\_essai

Lancement depuis le Workbench

Il suffit de cliquer deux fois sur l'icône du programme ou sur celle d'un fichier possédant le nom du programme dans son champ TOOL-TYPES. Dans le premier cas vous pouvez utiliser la touche SHIFT pour sélectionner autant de fichiers que vous le désirez avant de cliquer sur l'icône du programme.

L'icône du programme admet six types d'outils:

```
HELPPFILE ,  
STARTUP   ,  
LIBS      ,  
CLIPS     ,  
X_ICON    ,  
Y_ICON    ,
```

Utilisez le menu Information du Workbench pour les modifier.

À noter que les fenêtres peuvent être réduites ("icônifiées"), elles

perdent alors tous ces boutons, y compris ceux permettant leur fermeture, leur changement de plan ou leur changement de taille (zoom) est alors impossible.

## 1.6 Les divers objets manipulés par AmiCAD

Les objets manipulés par AmiCAD peuvent être de l'un des types suivants :

Lignes  
Connexions  
Composants  
Références  
Valeurs/Types  
Labels  
Connecteurs  
Textes  
Boîtes  
Ellipses  
Triangles  
Arcs de cercles  
Groupes

Chacun de ces objets peut être placé manuellement, en utilisant le menu Dessin approprié. Ils peuvent également être placés à l'aide d'une fonction ARexx, en utilisant un script par exemple.

Leur modification peut être faite en ouvrant une requête, en cliquant deux fois dessus à l'aide du bouton gauche de la souris (requête gtlayout.library). La plupart de ces requêtes permettent d'appliquer les changements (pour tous ou partie) soit au seul élément ayant été sélectionné pour la requête, soit à plusieurs objets identiques ou de même type, en utilisant le bouton Application. Pour certaines requêtes les boutons Étendre doivent être marqués pour que l'application ait lieu pour d'autres objets.

Ils peuvent être déplacés en cliquant dessus avec le bouton gauche de la souris, puis en déplaçant celle-ci tout en maintenant ce bouton appuyé. Il suffit de sélectionner plusieurs objets pour les déplacer en même temps (en utilisant la touche SHIFT, le menu Edition/Multisélection, ou en encadrant une zone à l'aide de la souris).

## 1.7 Les lignes

Les lignes tracées par AmiCAD peuvent être de plusieurs types :

- lignes simples (fils de liaisons)
  - lignes doubles (fils utilisés pour les schémas de puissance)
  - bus (tracés en traits forts)
  - lignes spéciales (tracés de décors, de largeur quelconque)
  - les lignes tracées en pointillés (contours de zones par exemple)
-

Seules les lignes simples (fils de liaison) sont reconnues par les scripts élaborant les "netlists".

Utilisez les menus Dessin/Placer fil à Dessin/Largeur quelconque ou bien les macros DRAW et DRAWMODE pour tracer des lignes.

Tous les types de lignes peuvent être associés à un label.

Pour modifier une ligne vous pouvez cliquer sur l'une de ses extrémités et faire glisser ce point à l'aide de la souris en maintenant le bouton appuyé. Vous pouvez également cliquer deux fois dessus à l'aide du bouton gauche et utiliser la requête qui s'ouvre alors. Celle-ci permet de modifier les paramètres suivants :

- les coordonnées des extrémités
- le type de ligne (pour les lignes spéciales)
- sa largeur
- le label associé à cette ligne.

## 1.8 Les connexions

Les connexions permettent de réaliser des interconnexions entre les lignes (deux lignes qui se croisent sans jonction ne sont normalement pas reliées). Elles peuvent être rondes ou carrées (en utilisant la forme alternée).

Attention certains scripts élaborant les "netlists" ne prennent pas en compte la forme alternée.

Pour les placer, vous pouvez utiliser le menu Dessin/Placer connexion ou la macro JUNCTION.

Elles peuvent être modifiées en cliquant deux fois dessus, la requête permet alors de changer les paramètres suivants :

- l'emplacement
- la taille horizontale (ou verticale)
- le choix du symbole (forme normale, ronde ou alternée, carrée)

## 1.9 Composants

Les composants sont des objets complexes, inclus dans des fichiers bibliothèques. Chacune de ces bibliothèques contient une famille de symboles différents.

Les bibliothèques de composants sont normalement situées dans le répertoire "Bibliothèques" du tiroir AmiCAD. Cependant ce chemin peut être modifié à l'aide du type d'outil LIBS, au lancement du programme ou à l'aide de la fonction ARexx LIBSPATH.

Les composants inclus dans les bibliothèques sont pratiquement tous définis sur une grille 10x10. Les connexions sont toutes sur cette grille. Un double clic du bouton gauche sur un composant

---



entraîne l'ouverture d'une requête permettant sa modification. Son emplacement, son échelle (horizontale et/ou verticale), la largeur de son tracé, une symétrie horizontale, une rotation par pas de 90 degrés, le choix d'un symbole alterné (s'il existe), le choix de la porte ou du circuit s'il y en a plusieurs, ainsi que la numérotation des bornes (visibles ou non) peuvent ainsi être édités facilement.

La référence et la valeur d'un composant peuvent aussi être définis à l'aide de cette requête. Ces deux éléments peuvent être ensuite placés où bon vous semble.

Il n'existe pour l'heure aucun éditeur de composant. Utilisez les clips si vous voulez utiliser un composant spécifique (de préférence en groupant tous les éléments).

## 1.10 Les Références affectées aux composants

Tout composant peut recevoir deux attributs: une référence et une valeur (ou un type, pour un circuit intégré par exemple).

La référence peut être placée en même temps que le composant ou bien d'abord placée en temps que texte puis liée à un composant existant (en utilisant la fonction LINKREF).

Leur édition peut être faite de façon autonome ou en ouvrant la requête pour le composant. Cet objet possède les mêmes attributs que les objets textes.

## 1.11 Les valeurs et les types affectés aux composants

Tout composant peut recevoir deux attributs: une référence et une valeur (ou un type, pour un circuit intégré par exemple).

La valeur (ou le type) peut être placé en même temps que le composant ou bien d'abord placé en temps que texte puis lié à un composant existant (en utilisant la fonction LINKVAL).

Leur édition peut être faite de façon autonome ou en ouvrant la requête pour le composant. Cet objet possède les mêmes attributs que les objets textes.

## 1.12 Les labels

Les labels sont utilisés en association avec une ligne définissant un fil, pour réaliser des "netlists". Ce sont des textes possédant un attribut particulier.

Pour les placer, vous pouvez utiliser l'une des deux méthodes qui suivent :

- placer d'abord le fil, éditer le fil et définir alors son

label

- placer le fil et un texte, puis sélectionner ces deux éléments et utiliser une macro du style `LINKLAB (FIRSTSEL, NEXTSEL (FIRSTSEL))`

Ils peuvent ensuite être modifiés en éditant soit le fil, soit le label lui-même (comme un texte).

Quand vous cliquez sur un fil possédant un label, celui-ci est aussi sélectionné. Mais l'inverse (clic sur un label en premier) n'entraîne pas la sélection du fil, ceci permet de placer le label où bon vous semble (il n'est cependant pas recommandé de le placer trop loin !).

## 1.13 Les connecteurs

Les connecteurs sont utilisés pour définir des entrées ou des sorties sur le schéma. Il existe d'ailleurs deux types de connecteurs (entrées ou sorties). Ils possèdent un point d'ancrage placé sur l'un de leurs côtés (à droite pour une entrée, à gauche pour une sortie, quand ils n'ont pas été tournés). Le sens de la flèche peut être changé en leur faisant subir une symétrie (après un double clic, dans la requête d'édition). Si vous sélectionnez un connecteur, puis utilisez le menu Dessin/Symétrie le connecteur sélectionné changera de type (entrée en sortie, sortie en entrée).

Utilisez le menu Dessin/Placer connecteur entrée, le menu Dessin/Placer connecteur de sortie ou l'une des fonctions INPUT ou OUTPUT pour placer ces objets.

Il est possible de les modifier en cliquant deux fois dessus, une requête est alors ouverte, permettant de modifier l'un des éléments suivants :

- emplacement
- les échelles verticale et horizontale
- la largeur du tracé
- la symétrie horizontale (sens de la flèche)
- leur orientation (rotations par pas de 90 degrés)
- la chaîne de caractères les définissant

## 1.14 Les textes

Les textes servent à apporter diverses précisions sur les dessins. Ils peuvent être liés aux composants pour définir leurs valeurs et leurs références, ou bien liés aux tracés pour définir des labels (utiliser les fonction LINKVAL, LINREF ou LINKLAB pour lier un texte existant à un composant ou une ligne déjà existante).

Ils peuvent être placés en utilisant le menu Dessin/Placer texte ou la fonction WRITE.

---

Il est aussi possible de les modifier en cliquant deux fois dessus, une requête est alors ouverte, permettant de modifier l'un des éléments suivants :

- emplacement
- les échelles verticale et horizontale
- la largeur du tracé
- symétrie horizontale
- leur orientation (rotations par pas de 90 degrés)
- la chaîne de caractères les définissant

## 1.15 Les boîtes (rectangles)

Les boîtes permettent de tracer divers décors ou bien des composants particuliers. Elles peuvent être vides (le contour seul est alors tracé) ou pleines.

Leur tracé s'effectue à l'aide du menu Dessin/Placer rectangle ou à l'aide de la fonction BOX.

Pour les modifier, il est possible de cliquer dans l'un des coins tout en maintenant le bouton gauche de la souris enfoncé, puis de déplacer ce coin à un autre endroit (utilisez le bouton droit pour annuler l'opération). À noter qu'en maintenant la touche CTRL enfoncée pendant cette opération, un carré est dessiné.

Il est aussi possible de les modifier en cliquant deux fois dessus, une requête est alors ouverte, permettant de modifier l'un des éléments suivants :

- coordonnées des coins
- largeur du tracé (rectangles vides)
- la nature de leur surface (vides ou pleins)

## 1.16 Les ellipses

Les ellipses permettent de tracer divers décors ou bien des composants particuliers. Elles peuvent être vides (le contour seul est alors tracé) ou pleines.

Leur tracé s'effectue à l'aide du menu Dessin/Placer ellipse ou à l'aide de la fonction ELLIPSE.

Pour les modifier, il est possible de cliquer sur l'un des bords (axe horizontal ou vertical) tout en maintenant le bouton gauche de la souris enfoncé, de déplacer ce coin à un autre endroit (utilisez le bouton droit pour annuler l'opération). À noter qu'en maintenant la touche CTRL enfoncée pendant cette opération c'est un cercle qui sera dessiné.

Il est aussi possible de les modifier en cliquant deux fois dessus, une requête est alors ouverte, permettant de modifier l'un des éléments suivants :

---

- emplacement
- rayons horizontal et vertical
- largeur du tracé (ellipses vides)
- la nature de leur surface (vides ou pleines)

## 1.17 Les triangles

Les triangles permettent de tracer divers décors ou bien des composants particuliers. Ils peuvent être vides (le contour seul est alors tracé) ou pleins.

Leur tracé s'effectue à l'aide du menu Dessin/Placer triangle ou à l'aide de la fonction TRIANGLE.

Pour les modifier, il est possible de cliquer dans l'un des coins tout en maintenant le bouton gauche de la souris enfoncé, de déplacer ce coin à un autre endroit (utilisez le bouton droit pour annuler l'opération). Il est aussi possible de les modifier en cliquant deux fois dessus, une requête est alors ouverte, permettant de modifier l'un des éléments suivants :

- coordonnées des coins
- largeur du tracé (triangles vides)
- la nature de leur surface (vides ou pleins)

## 1.18 Arcs de cercles

Les arcs de cercles peuvent être placés en utilisant le menu Dessin/Placer arc de cercle. Ils peuvent servir à l'élaboration de diverses figures (boîtes aux coins arrondis par exemple). Les rayons horizontaux et verticaux peuvent être différents ce qui en fait plutôt des arcs "d'ellipse".

Si un arc est sélectionné, les touches fléchées peuvent être utilisées de la façon suivante pour le modifier :

- sans aucune autre touche : déplacement pixel par pixel
- avec une touche SHIFT : déplacement par pas de la grille
- avec une touche ALT : modification des rayons pixel par pixel
- avec SHIFT-ALT : modification des rayons par pas de la grille
- avec la touche CRL : modification des angles de début ou de fin
- avec CTRL-SHIFT : modification des angles début/fin par pas.

Les arcs peuvent aussi être édités en effectuant un double clic à l'aide d'un double clic du bouton gauche de la souris (leur emplacement, les angles de début et de fin, et leur largeur peuvent alors être modifiés). Les angles sont donnés à 1 degré près.

La largeur des arcs peut être quelconque (mais pas de tracé en pointillés possible). La requête de modification permet de tracer des arcs de largeur égale à 10 pixels. Utilisez la fonction PENWIDTH pour une largeur supérieure.

La fonction ARC permet de placer des arcs de cercles.

## 1.19 Les groupes

Les groupes sont des ensembles d'objets (de type quelconque, lignes, jonctions, textes, labels, etc) ayant été sélectionnés puis regroupés dans le même sous-ensemble. Ils permettent facilement de créer des composants spéciaux, cependant ceux-ci ne pourront être utilisés pour élaborer des netlists par exemple.

Les groupes souvent utilisés peuvent être sauvés dans des fichiers, en utilisant les clips.

## 1.20 Barre de titre

La barre de titre des fenêtres de schéma comportent certaines indications:

- le nom (complet ou non) du fichier associé au schéma,
- quelques mentions entre crochets []:
- le signe + si le schéma a été modifié et n'a pas été sauvé,
- la lettre G si l'alignement sur la Grille est valide,
- la lettre R si les composants sont placés avec leur Référence,
- la lettre V si les composants sont placés avec leur Valeur,
- la lettre N si les composants sont placés avec leurs Numéros de broches,
- la lettre L si les Lignes sont tirées lors du déplacement de composants,
- un nombre indiquant la largeur des tracés en vigueur (0 si pointillés).
- l'abscisse du curseur X=...,
- une virgule,
- l'ordonnée du curseur Y=....,
- le rapport de zoom Z=...%

## 1.21 Type d'outil HELPFILE

Ce type d'outil permet de déterminer où se situe le fichier d'aide. Il doit être suivi du signe = puis du nom du fichier d'aide, avec son chemin complet. Vous pouvez ainsi placer ce fichier où vous le désirez, et même le renommer (cependant non conseillé !) puisque vous devez donner le nom complet du fichier.

Exemple :

HELPFILE=HELP:AmiCAD.guide

## 1.22 Type d'outil WINDOW

Ce type d'outil a été supprimé dans la version 2.05 (les dimensions du document au démarrage sont maintenant sauvées dans le fichier AmiCAD.prefs).

## 1.23 Type d'outil STARTUP

Ce type d'outil permet de lancer l'exécution d'un script ARexx dès le démarrage du programme, ceci vous permet par exemple de définir automatiquement les fonctions que vous souhaitez.

Ce type doit être suivi du signe = puis du nom du script avec ou sans l'extension amiCAD.

Exemples :

STARTUP=startup.amiCAD

STARTUP=démarrage

## 1.24 Type d'outil LIBS

Ce type d'outil permet de spécifier l'emplacement (le chemin) des bibliothèques de symboles. Par défaut, ce chemin est celui du répertoire nommé "Bibliothèques", situé dans le chemin courant du programme.

Exemples :

LIBS=Work:AmiCAD/Symboles

LIBS=Symboles

Voir aussi : LIBSPATH

## 1.25 Type d'outil CLIPS

Ce type d'outil permet de spécifier l'emplacement (le chemin) des clips définis par l'utilisateur. Par défaut, ce chemin est celui du répertoire nommé "Clips", situé dans le chemin courant du programme.

Exemples :

CLIPS=Work:AmiCAD/Clips

CLIPS=Symboles\_utilisateur

Voir aussi : CLIPPATH

## 1.26 Type d'outil MACRO

Cet outil a été supprimé dans la version 2.05.

## 1.27 Type d'outil X\_ICON

Ce type d'outil permet de déterminer l'abscisse des coordonnées d'affichage de l'icône du programme sur l'écran du Workbench. Il suffit d'ajouter la ligne X\_ICON=10 pour que l'icône soit affichée à gauche de l'écran du Workbench.

Voir aussi : Y\_ICON

## 1.28 Type d'outil Y\_ICON

Ce type d'outil permet de déterminer l'ordonnée des coordonnées d'affichage de l'icône du programme sur l'écran du Workbench. Il suffit d'ajouter la ligne Y\_ICON=500 pour que l'icône soit affichée en bas de l'écran du Workbench.

Voir aussi : X\_ICON

## 1.29 Type d'outil SHEET\_WIDTH

Ce type d'outil a été supprimé dans la version 2.05 (les dimensions du document au démarrage sont maintenant sauvées dans le fichier AmiCAD.prefs).

## 1.30 Type d'outil SHEET\_HEIGHT

Ce type d'outil a été supprimé dans la version 2.05 (les dimensions du document au démarrage sont maintenant sauvées dans le fichier AmiCAD.prefs).

## 1.31 Type d'outil GRIDSIZE

Ce type d'outil a été supprimé dans la version 2.05 (les dimensions de la grille au démarrage sont maintenant sauvées dans le fichier AmiCAD.prefs).

## 1.32 Les menus AmiCAD

Le programme AmiCAD possède six menus, pouvant être sélectionnés de façon classique, à l'aide du bouton droit de la souris. La plupart d'entre eux peut être également appelée à l'aide de raccourcis clavier, sans même utiliser la touche Amiga-droite.

Menu Projet  
Menu Dessin  
Menu Édition

---

Menu Scripts  
Menu Macros  
Menu Préférences

À noter que la plupart de ces menus peuvent être appelés en utilisant la fonction ARexx MENU.

### 1.33 Menu Projet

Ce menu possède 16 entrées:

Charger	Charge un schéma dans la fenêtre active.
Sauver	Sauve le schéma dans le fichier courant.
Sauver en	Sauve le schéma en spécifiant un nom de fichier.
Sauver format IFF	Sauve le schéma dans un fichier au format IFF.
Renommer	Pour renommer la fenêtre active.
Annoter~fichier	Pour annoter un fichier.
Supprimer~fichier	Permet de supprimer un fichier.
Réduire~fenêtre	Réduit la fenêtre active (icône).
Cacher~fenêtre	Ferme la fenêtre active, le texte reste en mémoire.
Autre~fenêtre	Ouvre une nouvelle fenêtre.
Initialiser	Vide la fenêtre courante.
Imprimer active	Impression du document contenu dans la fenêtre ←
Informations	Affiche quelques informations.
Schémas	Donne la liste des documents en mémoire.
Aide~	Affiche une aide AmigaGuide.
Quitter	Ferme tous les documents.

### 1.34 Menu Projet/Charger

Ce menu permet de choisir le schéma à charger dans la fenêtre active. La boîte de requête de fichiers utilise la bibliothèque asl.library (celle-ci est sur ~disque). À noter que celle-ci n'est chargée que lors de l'appel d'une des fonctions qui en a besoin (Chargement, Sauver en...). La fenêtre prend le nom du fichier chargé.

Si la fenêtre contenait un schéma, celui-ci est perdu, sauf s'il avait été modifié mais non sauvé: le programme vous offre alors le choix de continuer l'opération de chargement (l'ancien schéma est alors perdu) ou bien d'abandonner, vous permettant alors de le sauver.

Raccourci clavier : AMIGA-O (open) ou O

Nota : vous pouvez également utiliser les fonctions du système 3.0 pour charger un fichier: voir AppIcon.

Voir aussi : fonctions LOAD, OPEN

Appel ARexx : MENU("Charger")

---



### 1.35 Menu Projet/Sauver

Permet de sauvegarder le schéma dans un fichier disque en utilisant le nom de la fenêtre.

Raccourci clavier : AMIGA-S (sauver) ou S

Voir aussi : fonction SAVE

Appel ARexx: MENU("Sauver")

### 1.36 Menu Projet/Sauver en

Ce menu permet de sauvegarder le texte en choisissant le fichier destination à l'aide d'une boîte de requête. Pour ce menu vous ne pouvez pas cliquer deux fois sur le nom d'un fichier pour le sélectionner (obligation d'utiliser le gadget VALIDE pour éviter les erreurs)

Si le nom de fichier choisi correspond à un fichier déjà existant le programme vous demandera une confirmation avant de l'écraser.

Raccourci clavier : AMIGA-A (save As) ou A

Voir aussi : fonction SAVE

Appel ARexx: MENU("Sauver~en")                      Attention: espace "solide"  
(ALT-ESPACE) entre Sauver et en.

### 1.37 Menu Projet/Sauver format IFF

Ce menu permet de sauvegarder la fenêtre courante dans un fichier au format IFF, récupérable par un programme de dessin bitmap (Personal Paint, Deluxe Paint ou autre...). La sauvegarde se fait en monochrome (traits noirs sur fond blanc). La grille n'est pas incluse dans le fichier, même si elle est présente à l'écran lors de l'appel du menu.

Le choix du fichier destination est fait à l'aide de la requête de l'asl.library.

Il n'y a pas de raccourci clavier pour ce menu.

Voir aussi : fonction SAVEIFF

Appel ARexx: MENU("Sauver~format")                      Attention: espace "solide"  
(ALT-ESPACE) entre Sauver et format.

### 1.38 Menu Projet/Renommer

---

Ce menu permet de renommer le fichier associé au schéma de la fenêtre en cours.  
Le choix s'effectue à l'aide d'une boîte de requête de fichier.

Raccourci clavier : AMIGA--= ou =

Voir aussi : fonction FILENAME

Appel ARexx: MENU("Renommer")

### 1.39 Menu Projet/Annoter fichier

Ce menu permet d'ajouter une note à un fichier choisi à l'aide de la requête de l'asl.library (effet analogue à celui de la commande FileNote du DOS).

Cette note sera affichée avec la commande List de l'AmigaDOS ou visualisée à l'aide du menu Icônes/Information du Workbench.

Il n'y a pas de raccourci clavier pour ce menu.

Appel ARexx : MENU("Annoter")

### 1.40 Menu Projet/Supprimer fichier

Ce menu permet, comme son nom l'indique, de supprimer un fichier...  
Le fichier est choisi à l'aide du requester (pas de double clic là non plus). Attention à ce que vous faites (il faut plus de temps pour créer un fichier que pour l'effacer...).

Il n'y a pas de raccourci clavier pour ce menu (danger...).

Appel ARexx : MENU("Supprimer~fichier")      Attention: espace "solide"  
(ALT-ESPACE) entre Supprimer et fichier.

### 1.41 Menu Projet/Réduire fenêtre

Ce menu permet de réduire les dimensions de la fenêtre active.  
Une petite fenêtre apparaît alors en haut de l'écran avec le nom du schéma seul.  
Vous pouvez sélectionner cette fenêtre puis utiliser le bouton droit de la souris. Cette fonction est bien utile quand on travaille sur de multiples fenêtres.

Vous pouvez déplacer cette fenêtre à n'importe quel emplacement de votre choix, quand ce menu sera rappelée ultérieurement, cette petite fenêtre reprendra cette place.

Pour fermer une fenêtre qui a été réduite sans la rouvrir, sélectionnez-la

puis effectuez la combinaison CTRL-Q au clavier.

Raccourci clavier : AMIGA-I (icône) I

Appel ARexx : MENU("Réduire")

## 1.42 Menu Projet/Cacher fenêtre

Ce menu permet de fermer la fenêtre active. Le document reste cependant en mémoire. Pour faire réapparaître cette fenêtre vous pouvez cliquer deux fois sur l'icône d'application du programme, située sur l'écran du Workbench, ou bien utiliser un double clic sur le bouton droit de la souris, si vous travaillez dans une autre fenêtre, une requête est alors affichée comportant une liste de boutons contenant chacun le nom des divers documents présents en mémoire. Cliquez alors sur le nom correspondant à celui avec lequel vous voulez travailler.

Raccourci clavier : AMIGA-\$ ou \$

Appel ARexx : MENU("Cacher")

## 1.43 Menu Projet/Autre fenêtre

Cette fonction permet l'ouverture d'une nouvelle fenêtre. Une requête de fichier permet de choisir le schéma à y charger (utilisez le gadget ANNULER si vous ne voulez rien y mettre, mais utilisez alors plutôt la touche F4, comme expliqué ci-dessous).

Raccourci clavier : touche de fonction F3.

La touche de fonction F4 permet la même chose mais sans provoquer l'ouverture de la requête de fichier.

Une nouvelle fenêtre peut être ouverte, en tapant sur F3 ou F4, même à partir d'une fenêtre iconifiée (cette dernière ne sera pas réouverte).

Voir aussi : fonctions NEWSHEET, OPEN

Appel ARexx: MENU("Autre")

## 1.44 Menu Projet/Initialiser

Ce menu sollicite la demande d'effacement du schéma présent dans la fenêtre (Attention...). Si le document a été modifié, vous pourrez cependant le sauver (répondre Oui à la seconde requête).

Il n'y a pas de raccourci clavier pour ce menu.

Appel ARexx : MENU("Initialiser")

---

## 1.45 Menu Projet/Imprimer

Ce menu permet d'envoyer le schéma à l'imprimante.  
Le programme utilise les Préférences sélectionnées par le système : pilote d'imprimante... Le schéma entier est imprimé.

Il vous sera demandé le facteur d'agrandissement : si vous répondez 100 un pixel à l'écran sera imprimé sur un point (100%), si vous répondez 200, un pixel occupera deux points : le schéma sera alors deux fois plus grand. C'est à vous d'essayer pour trouver la meilleure échelle possible, selon la résolution de votre imprimante et les dimensions de votre schéma.

ATTENTION : la valeur saisie doit être non nulle. Elle ne doit pas être trop importante également, surtout si votre schéma est de grande taille, une valeur de 300 à 500 convient généralement.

Il vous sera aussi demandé si vous voulez imprimer votre schéma avec ou sans rotation (Paysage ou Portrait) : ceci permet d'orienter le schéma dans le sens horizontal ou le sens vertical de la feuille.

Raccourci clavier : AMIGA-P (Print) ou P

Voir aussi : PRINT

Appel ARexx : MENU("Imprimer")

## 1.46 Menu Projet/Informations

Ce menu provoque l'affichage d'un certain nombre d'informations : copyright, nombre d'objets dans le schéma, nom du port ARexx, quantité de mémoire libre...

Raccourci clavier : AMIGA-K ou K

Appel ARexx : MENU("Informations")

## 1.47 Menu Projet/Schémas

Ce menu permet d'afficher dans une boîte le nom des différents schémas. Il suffit de cliquer sur le nom d'un schéma pour amener la fenêtre en avant et pouvoir y travailler.

Vous pouvez aussi effectuer un clic sur le bouton central de la souris pour appeler cette fonction.

Il n'y a pas de raccourci clavier pour ce menu.

Voir aussi : fonction REQSHEET.

Appel ARexx : MENU("Schémas").

---

## 1.48 Menu Projet/Aide

Ce menu permet d'obtenir une aide à l'aide du programme AmigaGuide. Vous devez pour cela spécifier le nom d'un "node" appartenant au fichier AmiCAD.guide. Ce fichier contient notamment des nodes pour chacune des macros ARexx (Ex: COPY, PASTE...)

À noter que vous pouvez également obtenir une aide à partir de chacune des entrées de menu en sélectionnant l'un de ces menus et en appuyant simultanément sur la touche HELP. D'autre part quand un message d'erreur concernant une fonction ARexx est affichée, appuyez sur la touche HELP alors que la requête est encore ouverte pour bénéficier de l'aide gérée par AmigaGuide pour cette fonction.

Raccourci clavier : AMIGA-? or ?

Voir aussi : fonction HELP.

Appel ARexx: MENU("Aide")

## 1.49 Menu Projet/Quitter

Ce menu permet de fermer toutes les fenêtres et de supprimer tous les schémas de la mémoire. Le programme quitte également en libérant la mémoire qu'il occupait.

Si l'un des schémas a été modifié, une sauvegarde sera proposée.

Raccourci clavier : AMIGA-Q (quitter)

Q (ne ferme que la fenêtre active)

Voir aussi : fonction CLOSE.

Appel Arexx: MENU("Quitter")      Quitte sans demander de sauvegarde

## 1.50 Icône d'application AmiCAD

Le concept d'AppIcon (Icône d'application) autorise le chargement d'un fichier de façon très simple, en une seule action, à l'aide de la souris. Comme décrit, dans l'introduction, le programme crée une icône qui apparaît sur l'écran du Workbench, lors de son lancement. Cette icône porte le nom d'AmiCAD.

Il vous suffit de "saisir" une icône correspondant à un fichier schéma quelconque, à l'aide de la souris, et de la faire glisser sur cette icône pour que ce fichier soit chargé dans une nouvelle fenêtre.

Cette icône possède également une autre fonction: en cliquant deux fois dessus rapidement, avec le bouton gauche, vous pouvez "réveiller" le programme si toutes les fenêtres sont cachées (la fenêtre active est réouverte), ou faire passer en avant-plan la fenêtre active si celle-ci est cachée. Cette action peut également permettre un déblocage éventuel, provoqué par une mauvaise utilisation de la

fonction LOCK dans un script ARexx.

Si vous cliquez sur cette icône alors qu'aucune tâche n'est libre, pour cause d'impression ou de requête en cours de traitement, le programme vous demandera si vous voulez ouvrir une nouvelle fenêtre.

## 1.51 Gestion de la mémoire

Les pools correspondent à un concept de gestion de la mémoire amélioré, permettant de ne pas trop fragmenter la mémoire. En effet chaque objet est mémorisé de façon dynamique, dans une zone mémoire qui est allouée au programme au fur et à mesure des besoins. Ceci a généralement pour conséquence de créer de nombreuses zones de mémoire libre et occupée, en alternance, dans la mémoire de votre Amiga. Grâce à ce concept, cet effet est beaucoup diminué, et surtout la vitesse est nettement améliorée. Malheureusement ce concept exige l'utilisation du système 3.0 au moins.

## 1.52 Menu Dessin

Ce menu est destiné à placer les objets sur le document. Tous ces objets sont des objets vectoriels, ils peuvent être modifiés en cliquant dessus deux fois avec le bouton gauche de la souris. Une requête contenant des boutons est alors ouverte, selon le type d'objet. Il suffit alors de cliquer sur le bouton approprié pour changer une des caractéristiques de l'objet choisi.

Ce menu possède 28 entrées:

- Pas~de~la~grille=~10
- Aligner~sur~grille
- Choix composant
- Placer~composant
  - Placer~référence
  - Placer~valeur/type
  - Placer~numéros~bornes
- Rotation
- Symétrie
- Alternner~symbole
- Position~normale
- Placer~fil
  - Tracé~orthogonal
  - Tracé~continu
  - Trait~double
  - Trait~fort~(Bus)
  - Largeur quelconque
  - Pointillés
- Placer rectangle
- Placer triangle
- Placer~ellipse
  - Surfaces pleines
- Placer~arc~de~cercle

Placer~connexion  
Placer~texte  
Placer~connecteur~entrée  
Placer~connecteur~sortie  
Redessiner~tout

### 1.53 Menu Dessin/Pas de la grille= 10

Ce menu détermine la taille de la grille permettant d'aligner plus facilement les éléments et leurs connexions. Cette grille possède une taille par défaut de 10 pixels, les composants définis dans les diverses bibliothèques de symboles tiennent tous compte de cette dimension. Vous pouvez cependant travailler avec une taille différente si vous le souhaitez.

Raccourci clavier : AMIGA- $\mu$  ou  $\mu$

Voir aussi : macro ARexx SETGRID.

Appel ARexx: MENU("Pas~grille")      Attention: espace "solide"  
(ALT-ESPACE) entre Pas et grille.

### 1.54 Menu Dessin/Aligner sur grille

Ce menu, quand il est marqué, permet de placer les objets (composants, fils, jonctions, etc...) directement aux intersections de la grille. Ceci permet de travailler plus vite, tout en alignant plus facilement les éléments qui doivent être connectés.

Raccourci clavier: AMIGA-G ou G

Voir aussi : macro ARexx SNAPGRID.

Appel ARexx: MENU("Aligner")

### 1.55 Menu Dessin/Choisir composant

Ce menu permet de choisir un composant (ou un symbole) dans une bibliothèque. Attention: ce menu fait appel à la gtlayout.library. Une fenêtre comportant deux listes est ouverte: la liste de gauche comporte les noms des bibliothèques déjà chargées, la liste de droite comporte les noms des symboles appartenant à la bibliothèque sélectionnée.

Vous pouvez charger ou supprimer une bibliothèque en utilisant les boutons correspondants. Les boutons "Placer avant" et "Placer après" permettent de changer l'ordre des bibliothèques (les symboles seront cherchés d'abord dans la première, puis dans les suivantes...) Les boutons situés en bas à gauche sont équivalents aux menus correspondants.

---

À noter que le composant sélectionné est affiché dans la fenêtre de travail, en haut à gauche. Vous pouvez cliquer deux fois de suite sur un nom pour choisir un symbole ou utiliser le bouton "Accepter". Le bouton "Trier" permet de classer les symboles de la liste dans l'ordre alphabétique.

Raccourci clavier: AMIGA-% ou %

Voir aussi : macros ARexx PUTPART, LOADLIB.

Appel ARexx: MENU("Choisir~composant")      Attention: espace "solide"  
(ALT-ESPACE) entre Choisir et composant.

## 1.56 Menu Dessin/Placer composant

Ce menu permet de placer un composant ou un symbole sur la feuille de dessin. Si aucun objet n'a été choisi, la requête permettant d'effectuer ce choix est ouverte.

Quand un objet valide a été choisi, le curseur se transforme, en dessinant cet objet. Il suffit alors de déplacer le curseur à l'emplacement souhaité puis de cliquer avec le bouton gauche de la souris pour le fixer.

Raccourci clavier: AMIGA-H ou H

Voir aussi : fonctions PUTPART

Appel ARexx: MENU("Placer~composant")      Attention: espace "solide"  
(ALT-ESPACE) entre Placer et composant.

## 1.57 Menu Dessin/Placer référence

Quand ce menu est marqué, lors du placement d'un composant, sa référence (par défaut) sera automatiquement placée sur le schéma, à proximité. À noter que cette référence, qui est un objet du type texte, peut être déplacée à votre convenance, simplement en cliquant dessus à l'aide du bouton gauche de la souris, puis tout en maintenant celui-ci appuyé, de la déplacer à l'emplacement souhaité et de le relâcher (vous pouvez d'ailleurs temporairement désactiver l'option Aligner sur grille pour mieux ajuster sa place). Les références par défaut sont définies dans les bibliothèques de symboles (Exemple D pour une diode, R pour une résistance...) Pour modifier cette référence ou la compléter, vous pouvez cliquer deux fois sur le composant ou la référence elle-même, puis choisir l'option "Modifier texte" dans la requête qui suit.

Vous pouvez utiliser le script ARexx "Référencer" pour numéroter les composants présents sur le document.

Il n'y a pas de raccourci clavier pour ce menu.



Voir aussi : macro ARexx SETREF.

Pas d'appel ARexx possible pour ce menu.

## 1.58 Menu Dessin/Placer valeur/type

Quand ce menu est marqué, lors du placement d'un composant, sa valeur (par défaut) sera automatiquement placée sur le schéma, à proximité. À noter que cette valeur, qui est un objet du type texte, peut être déplacée à votre convenance, simplement en cliquant dessus à l'aide du bouton gauche de la souris, puis tout en maintenant celui-ci appuyé, de la déplacer à l'emplacement souhaité et de le relâcher.

Les valeurs par défaut prennent le nom du composant défini dans la bibliothèque de symboles auquel il appartient. Pour modifier cette valeur ou la compléter, vous pouvez cliquer deux fois sur le composant ou la valeur elle-même, puis choisir l'option "Modifier texte" dans la requête qui suit.

Il n'y a pas de raccourci clavier pour ce menu.

Voir aussi : macro ARexx SETVAL.

Pas d'appel ARexx possible pour ce menu.

## 1.59 Menu Dessin/Placer numéros bornes

Quand ce menu est marqué, lors du placement d'un composant, les numéros de ses bornes seront indiqués, s'il y a lieu (pour un circuit intégré par exemple).

Quand un composant est déjà placé sur le schéma, cette indication peut être modifiée en cliquant deux fois dessus, puis en choisissant l'option "Numérotation bornes". À noter que cette option n'apparaîtra que s'il y a lieu de le faire (les bornes d'une résistance ne sont par exemple jamais marquées).

Il n'y a pas de raccourci clavier pour ce menu.

Voir aussi : SETPINS.

Pas d'appel ARexx possible pour ce menu.

## 1.60 Menu Dessin/Rotation

Ce menu permet de faire tourner d'un quart de tour le composant courant (en cours de placement, sous le curseur) ou le bloc d'objets sélectionné.

Il s'applique également sur un clip qui est en cours de positionnement (voir menu {"Édition/Coller" LINK "Paste~from~clip"}).

La rotation s'effectue dans le sens des aiguilles d'une montre, sauf si vous pressez une des touches SHIFT lors de l'appel du menu.

Raccourci clavier : AMIGA-R ou R (sens "normal")  
AMIGA-SHIFT-R ou SHIFT-R (sens inverse)

Voir aussi : macro ARexx ROTATE.

Appel ARexx: MENU("Rotation")

## 1.61 Menu Dessin/Symétrie

Ce menu permet de dessiner l'objet courant ou les objets sélectionnés en effectuant une symétrie par rapport à leur axe vertical (ou horizontal s'il ont été tournés).  
Il s'applique également sur un clip qui est en cours de positionnement (voir menu {"Édition/Coller" LINK "Paste~from~clip"}).

Raccourci clavier : AMIGA-/ ou /

Voir aussi : fonction ARexx SYMMETRY.

Appel ARexx: MENU("Symétrie").

## 1.62 Menu Dessin/Alterner symbole

Ce menu permet de choisir un second symbole pour un composant, quand il existe (défini dans les fichiers bibliothèques).  
Quand le curseur est en mode "placement composant", c'est lui qui est modifié, le composant éventuellement placé ensuite (avec un clic du bouton gauche), aura donc ce symbole.  
Sinon si certains éléments sont sélectionnés lors de l'appel de ce menu, leurs symboles seront changés.

Cette fonction est utile pour dessiner un ampli avec les entrées + et - inversées par exemple. Dans certaines bibliothèques les symboles alternés sont des symboles américains ou aux anciennes normes (Résistance par exemple).

À partir de la version 1.5, ce menu s'applique aussi aux jonctions, celles-ci peuvent être de forme ronde ou carrée.

À partir de la version 1.6, ce menu s'applique aussi aux connecteurs: la flèche change alors de direction.

Raccourci clavier : AMIGA-~ ou ~

Voir aussi : fonction CONVERT.

Appel ARexx: MENU("Alterner")

---

## 1.63 Menu Dessin/Position normale

L'appel de ce menu provoque l'annulation des effets obtenus par les précédents: rotation, symétrie et symbole alterné. Le curseur ou les composants sélectionnés sont donc amenés à leur position initiale (aucune rotation, pas de symétrie, utilisation du symbole normal).

Raccourci clavier : AMIGA-N ou N.

Appel ARexx : MENU("Position~normale")      Attention: espace "solide"  
(ALT-ESPACE) entre Position et normale.

## 1.64 Menu Dessin/Placer fil

L'appel de ce menu fait passer le curseur en mode "placement de fil". Le curseur se transforme alors pour indiquer ce mode (il prend la forme d'une croix). Pour placer un fil, il suffit ensuite de cliquer sur son point de départ, puis de recliquer sur son point d'arrivée. Pour interrompre ce mode, appuyez sur le bouton droit de la souris.

La largeur des lignes dépend du mode alors valide, reportez-vous aux menus trait double, trait fort, pointillés ou Largeur quelconque. Cette largeur est indiquée dans la barre de titre.

Pour modifier le tracé d'un fil, cliquez sur un de ses points de départ ou d'arrivée, puis tout en maintenant le bouton gauche de la souris appuyé, déplacez celui-ci sur le point d'arrivée souhaité, et relâchez le bouton. Vous pouvez appuyer sur le bouton droit avant de relâcher le bouton gauche si vous désirez annuler l'opération.

Vous pouvez également utiliser le clavier pour ajuster le placement d'une ligne: utilisez les touches fléchées pour les déplacer. Si vous utilisez conjointement les touches CTRL ou ALT, seul le début ou la fin de la ligne sera déplacé, si vous utilisez aussi une touche SHIFT le déplacement sera plus rapide (égal au pas de la grille).

Raccourci clavier : AMIGA-ESPACE ou ESPACE

Voir aussi : tracé orthogonal, tracé continu, trait double, trait fort et pointillés.

Fonction ARexx DRAW.

Appel ARexx: MENU("Placer~fil")      Attention: espace "solide"  
(ALT-ESPACE) entre Placer et fil.

## 1.65 Menu Dessin/Tracé orthogonal

Quand ce menu est marqué, les fils ne peuvent être tracés que verticalement, horizontalement ou bien à 45\textdegree{}.

Sélectionnez ce menu pour changer son état.

---

Raccourci clavier : AMIGA-| ou |

Voir aussi : fonction ARexx SETORTHO.

Appel ARexx: MENU("Tracé~orthogonal")      Attention: espace "solide"  
(ALT-ESPACE) entre tracé et orthogonal.

## 1.66 Menu Dessin/Tracé continu

Quand ce menu est marqué, lors d'une opération de placement de fils, dès que vous placez un fil, le point d'arrivée du dernier fil qui a été placé devient le point de départ du fil suivant. Appuyez sur le bouton droit de la souris pour interrompre la continuité du tracé, et recommencer ailleurs. Vous pouvez également tracer une ligne de longueur nulle, c'est à dire possédant les points de départ et d'arrivée confondus, la ligne ne sera alors pas tracée et vous pourrez commencer une nouvelle ligne à n'importe quel autre endroit.

Il n'y a pas de raccourci clavier pour ce menu.

Pas d'appel ARexx possible pour ce menu.

## 1.67 Menu Dessin/Trait double

Quand ce menu est marqué, les fils sont tracés en double largeur, leur largeur est alors de deux pixels à l'écran.

Pour annuler ce mode, sélectionnez-le à nouveau, ou bien choisissez l'un des deux suivants (Trait fort, ou Pointillés). Si aucun de ces modes n'est marqué le tracé s'effectue avec une largeur de un pixel.

Raccourci clavier : AMIGA-\ ou \

Voir aussi : fonctions ARexx DRAWMODE(2), DRAW.

Appel ARexx: MENU("Trait~double")      Attention: espace "solide"  
(ALT-ESPACE) entre Trait et double.

## 1.68 Menu Dessin/Trait fort (Bus)

Quand ce menu est marqué, les fils sont tracés en quadruple largeur, leur largeur est alors de cinq à sept pixels à l'écran (selon leur orientation).

Pour annuler ce mode, sélectionnez-le à nouveau, ou bien choisissez l'un des deux autres (Trait double, ou Pointillés). Si aucun de ces modes n'est marqué le tracé s'effectue avec une largeur de un pixel.

---

Raccourci clavier : Amiga-. ou .

Voir aussi : fonctions ARExx DRAWMODE(3), DRAW.

Appel ARExx: MENU("Trait~fort")      Attention: espace "solide"  
(ALT-ESPACE) entre Trait et fort.

## 1.69 Menu Dessin/Largeur quelconque

Quand ce menu est marqué, les lignes peuvent être tracées de n'importe quelle largeur, le programme vous demande de quelle largeur doivent alors être tracées les lignes. Cette fonction est valide pour tous les objets qui seront ensuite placés sur la feuille de dessin.

Il n'y a pas de raccourci clavier pour ce menu.

Voir aussi : fonctions ARExx DRAWMODE(-X), DRAW.

Pas d'appel ARExx possible pour ce menu.

## 1.70 Menu Dessin/Pointillés

Quand ce menu est marqué, les fils sont tracés en pointillés, en simple largeur.

Pour annuler ce mode, sélectionnez-le à nouveau, ou bien choisissez l'un des deux autres (Trait double, ou Trait fort).

Si aucun de ces modes n'est marqué le tracé s'effectue avec une largeur de un pixel.

Raccourci clavier : AMIGA-: ou :

Voir aussi : fonctions ARExx DRAWMODE (0), DRAW.

Appel ARExx: MENU("Pointillés")

## 1.71 Menu Dessin/Placer rectangle

La sélection de ce menu permet de tracer un rectangle. Le curseur prend la forme qu'il a également lors du tracé des lignes. Cliquez à l'emplacement désiré pour le premier coin du rectangle puis déplacez le pointeur à l'endroit désiré pour l'autre coin, cliquez à nouveau (avec le bouton gauche, naturellement).

Les rectangles peuvent prendre n'importe quelle largeur de trait, comme indiqué par les menus précédents.

Pour modifier le tracé d'un rectangle ainsi obtenu, cliquez dans l'un de ses coins et, tout en maintenant le bouton gauche appuyé, déplacez la souris jusqu'à ce que la position désirée soit atteinte. Relâchez alors le bouton. Vous pouvez appuyer sur

le bouton droit avant de relâcher le bouton gauche si vous désirez annuler l'opération.

Pour modifier la largeur du tracé d'un rectangle déjà placé, cliquez deux fois dessus avec la souris puis tapez la largeur souhaitée dans la boîte qui s'ouvre alors.

Pour tracer un carré, maintenez la touche CTRL appuyée lors du tracé.

Raccourci clavier : AMIGA-B ou B

Voir aussi : fonctions BOX, DRAWMODE, SETFILL.

Appel ARexx: MENU("Placer~rectangle")      Attention: espace "solide"  
(ALT-ESPACE) entre Placer et rectangle.

## 1.72 Menu Dessin/Placer triangle

La sélection de ce menu permet de tracer un triangle, si le menu Surfaces pleines est alors marqué, il sera plein.

Les triangles peuvent prendre n'importe quelle largeur de trait, comme indiqué par les menus précédents.

Pour modifier la largeur du tracé d'un triangle déjà placé, cliquez deux fois dessus avec la souris puis tapez la largeur souhaitée dans la boîte qui s'ouvre alors.

Raccourci clavier : AMIGA-' ou '

Voir aussi : fonction TRIANGLE, DRAWMODE, SETFILL.

Appel ARexx: MENU("Placer~triangle")      Attention: espace "solide"  
(ALT-ESPACE) entre Placer et triangle.

## 1.73 Menu Dessin/Placer ellipse

La sélection de ce menu fait passer le curseur en mode "placement d'ellipse". Une ellipse est alors dessinée à l'écran et celle-ci suit les mouvements de la souris. Pour placer une ellipse, cliquez alors sur le bouton gauche. Le diamètre peut être modifié en utilisant les touches fléchées (FLÈCHES verticales pour le rayon vertical, FLÈCHES horizontales pour le rayon horizontal) qui permet d'augmenter ou de diminuer ces rayons d'une dimension égale au pas de la grille. Quand une ellipse a été fixée sur le schéma, il est possible de la modifier en cliquant sur l'un de ses axes (vertical ou horizontal), près de sa périphérie, puis tout en maintenant le bouton de la souris appuyé, de déplacer celle-ci jusqu'à ce que la dimension souhaitée soit atteinte. Si vous souhaitez obtenir un cercle, maintenez la touche Ctrl enfoncée lors de ces opérations : les deux diamètres seront alors égaux.

---

Pour déplacer une ellipse, cliquez en son centre puis, tout en maintenant le bouton appuyé, faites glisser la souris jusqu'à ce que la position désirée soit atteinte.

Pour annuler ce mode utilisez le bouton droit de la souris, ou choisissez un autre mode.

Raccourci clavier : AMIGA-E ou E

Voir aussi : fonction ELLIPSE, surfaces pleines.

Appel ARexx: MENU("Placer~ellipse")                      Attention: espace "solide"  
(ALT-ESPACE) entre Placer et ellipse.

## 1.74 Menu Dessin/Surfaces pleines

Quand ce menu est marqué, les rectangles, les triangles et les ellipses qui seront alors tracés seront pleins, sinon seuls leurs contours seront tracés, en utilisant la largeur de trait alors sélectionnée.

Raccourci clavier : AMIGA-" ou "

Voir aussi : fonction ARexx SETFILL.

## 1.75 Menu Dessin/Placer arc de cercle

La sélection de ce menu fait passer le curseur en mode "placement d'arc". Ces arcs de cercle ont une précision de 1 degré maximum. Pour choisir les dimensions, utilisez les touches fléchées, avec ou sans SHIFT. La touche SHIFT permet d'avoir un changement plus rapide, égal au pas de la grille.

Pour placer un arc cliquez sur le bouton gauche de la souris quand la position désirée est atteinte.

Pour modifier le diamètre d'un arc déjà placé sur le schéma, cliquez deux fois dessus, puis saisissez les valeurs demandées, ou sélectionnez-le puis utilisez les touches fléchées en combinaison avec les touches SHIFT et ALT, comme lors du placement.

De la même façon, pour modifier les angles début ou fin, utilisez la touche CONTROL, avec ou sans touche SHIFT.

Pour déplacer un arc, cliquez vers son centre puis, tout en maintenant le bouton appuyé, faites glisser la souris jusqu'à ce que la position désirée soit atteinte.

Pour annuler ce mode utilisez le bouton droit de la souris, ou choisissez un autre mode de dessin.

Vous pouvez utiliser les commandes de rotation et/ou de symétrie pour changer rapidement les valeurs des angles.

---

Raccourci clavier : AMIGA-\$ ou \$

Voir aussi : fonction ARC.

Appel ARexx: MENU("Placer~arc")                      Attention: espace "solide"  
(ALT-ESPACE) entre Placer et arc.

## 1.76 Menu Dessin/Placer connexion

La sélection de ce menu fait passer le curseur en mode "placement de connexion".

Pour placer une connexion, cliquez sur le bouton gauche de la souris quand la position désirée est atteinte.

Pour annuler ce mode utilisez le bouton droit de la souris, ou choisissez un autre mode.

Raccourci clavier : AMIGA-J ou J

Voir aussi : fonction JUNCTION.

Appel ARexx: MENU("Placer~connexion")                      Attention: espace "solide"  
(ALT-ESPACE) entre Placer et connexion.

## 1.77 Menu Dessin/Placer texte

La sélection de ce menu entraîne l'apparition d'une requête, demandant quel est le texte à placer. Si le choix est positif, le curseur se transforme en texte, utilisez alors le bouton gauche de la souris pour placer celui-ci à l'emplacement désiré.

Pour modifier un texte, cliquez deux fois de suite sur celui-ci puis choisissez l'option proposée (Modifier texte, échelle verticale ou échelle horizontale).

Pour annuler ce mode utilisez le bouton droit de la souris, ou choisissez un autre mode.

Raccourci clavier : AMIGA-T ou T

Voir aussi : fonction WRITE.

Appel ARexx: MENU("Placer~texte")                      Attention: espace "solide"  
(ALT-ESPACE) entre Placer et texte.

## 1.78 Menu Dessin/Placer connecteur entrée

Permet de placer un symbole correspondant à une liaison avec un élément extérieur. Le point d'attache de cet objet est situé à sa droite. Il est dessiné à gauche du curseur. La taille de cet objet sera

---



automatiquement adaptée à son nom (un cadre est dessiné autour du texte, avec une flèche à sa gauche, en position "normale").

Comme pour un composant, cliquez à l'aide du bouton gauche de la souris pour fixer l'emplacement de cet élément, après lui avoir donné son nom dans la requête prévue à cet effet.

Pour modifier le nom d'un connecteur, cliquez deux fois dessus à l'aide du bouton gauche de la souris, puis choisissez l'option "Modifier texte".

Si vous voulez que la flèche soit dirigée vers la droite, utilisez le menu Dessin/Alternar~symbole.

Pour changer le type d'un connecteur (sortie en entrée, ou vice-versa), sélectionnez-le et utilisez le menu symétrie.

Pour annuler ce mode utilisez le bouton droit de la souris, ou choisissez un autre mode.

Raccourci clavier : AMIGA-< ou <

Voir aussi : fonction INPUT.

Appel ARexx: MENU("Placer~connecteur~entrée")    Attention: espaces "solides" (ALT-ESPACE) entre les mots.

## 1.79 Menu Dessin/Placer connecteur sortie

Permet de placer un symbole correspondant à une liaison avec un élément extérieur. Le point d'attache de cet objet est situé à sa gauche.

La taille de cet objet sera automatiquement adaptée à son nom (un cadre est dessiné autour du texte, avec une flèche à sa droite, en position "normale").

Comme pour un composant, cliquez à l'aide du bouton gauche de la souris pour fixer l'emplacement de cet élément, après lui avoir donné son nom dans la requête prévue à cet effet.

Pour modifier le nom d'un connecteur, cliquez deux fois dessus à l'aide du bouton gauche de la souris, puis choisissez l'option "Modifier texte".

Si vous voulez que la flèche soit dirigée vers la gauche, utilisez le menu Dessin/Alternar~symbole.

Pour changer le type d'un connecteur (sortie en entrée, ou vice-versa), sélectionnez-le et utilisez le menu symétrie.

Pour annuler ce mode utilisez le bouton droit de la souris, ou choisissez un autre mode.

Raccourci clavier : AMIGA-> ou >

Voir aussi : fonction OUTPUT.

Appel ARexx: MENU("Placer~connecteur~sortie")    Attention: espaces "solides" (ALT-ESPACE) entre les mots.

---

## 1.80 Menu Dessin/Redessiner tout

Permet d'effacer toute la feuille de schéma puis de la redessiner. Utile après un travail de sélection, de déplacement de blocs, ou pour réafficher correctement la grille.

Raccourci clavier : AMIGA-Z ou Z

Appel ARexx : MENU("Redessiner")

## 1.81 Les clips AmiCAD

Les clips sont des ensembles d'objets quelconques, ils sont définis simplement en sélectionnant des objets puis en utilisant les menus Édition/Copier ou Édition/Couper. Ils peuvent alors être collés n'importe où sur le document (ou même sur un autre document). Ils peuvent être également être sauves et relus ultérieurement.

À noter que les clips doivent respecter quelques règles simples si vous voulez qu'ils soient utilisables facilement. Ainsi si vous voulez que les connexions soient alignées sur la grille il est impératif que le point de référence du clip soit lui-même aligné sur cette grille: ce point est défini par le coin le plus haut à gauche du rectangle défini par les divers objets inclus dans ce clip. Bien sûr les connexions doivent se trouver à une distance de ce point qui soit un multiple du pas de la grille.

Voir aussi : CLIPUNIT

## 1.82 Menu Édition

Cette série de menus est relative aux blocs d'éléments sélectionnés. Avant de commencer leur description il faut expliquer comment sont sélectionnés ces blocs.

Cette sélection peut s'effectuer de deux manières :

- en cliquant dans une zone ne contenant aucun élément puis en maintenant le bouton gauche de la souris, et en faisant glisser celle-ci vous pouvez encadrer une zone dans laquelle tous les éléments présents seront sélectionnés,
- en cliquant sur un élément, celui-ci sera sélectionné. Cependant si le menu Édition/Multisélection n'est pas marqué, seul un élément peut être sélectionné, sauf si vous enfoncez une des touches SHIFT au moment où vous cliquez.

Quand un bloc est déjà sélectionné, vous pouvez l'étendre en pressant une des touches SHIFT avant de cliquer avec le bouton gauche de la souris, pour n'importe quelle des deux méthodes vues précédemment.

Pour annuler la sélection d'un élément ou d'un bloc, pressez la touche Ctrl en même temps que vous utilisez la souris (lors du glissement), tout en maintenant ou non la touche SHIFT appuyée afin de conserver la sélection des autres éléments déjà sélectionnés.

Pour sélectionner un élément situé en arrière-plan, sélectionnez d'abord celui qui est situé devant, puis recliquez en appuyant sur la touche ALT, avec ou sans la touche SHIFT (pour garder les autres sélections ou non).

Pour annuler la sélection d'un bloc pressez la touche ESC ou cliquez à nouveau deux fois de suite n'importe où sur l'écran.

Pour modifier les caractéristiques d'un objet, cliquez deux fois de suite dessus à l'aide du bouton gauche de la souris, puis choisissez les boutons correspondant aux caractéristiques que vous souhaitez modifier.

- Copier
- Coller
- Couper
- Effacer
- Cloner
- Fixer sur grille
- Passer devant
- Passer derrière
- Doubler taille
- Diviser taille
- Étirer
- Grouper
- Séparer
- Sauver clip
- Charger clip
- Multisélection
- Restaurer

### 1.83 Menu Édition/Copier

Ce menu permet la recopie du bloc spécifié dans une zone mémoire. Si l'opération réussit, la sélection est annulée. Le programme gère 10 zones mémoires différentes permettant ainsi de stocker 10 blocs différents (voir CLIPUNIT).

Raccourci clavier : AMIGA-C (copier) ou C

Voir aussi : fonctions COPY, PASTE.

Appel ARexx : MENU("Copy")

### 1.84 Menu Édition/Coller

Le contenu de la zone mémoire valide est copié à l'emplacement actuel du curseur. Il se déplace avec la souris, pour placer ce bloc, cliquez sur le bouton gauche. Seuls les éléments qui ne débordent pas de la fenêtre seront placés. Vous pouvez utiliser les menus permettant la rotation, la symétrie, l'agrandissement ou le retrécissement des objets sélectionnés pendant l'utilisation de ce menu.

Raccourci clavier : AMIGA-V ou V

Voir aussi : fonctions CLIPUNIT, PASTE, COPY.

Appel ARexx : MENU("Paste")

## 1.85 Menu Édition/Couper

Ce menu permet d'effacer le bloc spécifié (le bloc est copié dans la zone mémoire alors valide).

Attention : cette fonction peut modifier les numéros des objets restants.

Raccourci clavier : AMIGA-X ou X

Voir aussi : fonctions CLIPUNIT, PASTE, COPY.

Appel ARexx : MENU("Cut")

## 1.86 Menu Édition/Effacer

Ce menu permet d'effacer le bloc spécifié. Vous pouvez aussi utiliser la touche DEL.

Attention : cette fonction peut modifier les numéros des objets restants.

Raccourci clavier : AMIGA-Y, Y ou DEL

Appel ARexx : MENU("Delete")

## 1.87 Menu Édition/Cloner

Ce menu permet de recopier le bloc spécifié.

Raccourci clavier : AMIGA-TAB ou TAB

Appel ARexx : MENU("Clone")

## 1.88 Menu Édition/Fixer sur grille

Ce menu permet d'aligner les éléments appartenant au bloc spécifié sur la grille (leur point d'attache, déterminé par le point correspondant au pointeur de la souris, lors de leur pose, est fixé sur le point le plus proche de la grille).

Raccourci clavier : AMIGA-! ou !

Appel ARexx : MENU("Fix")

---

## 1.89 Menu Édition/Passer devant

Ce menu permet de faire passer un élément en tête de la liste, il est alors choisi s'il est en recoupement avec d'autres quand l'utilisateur clique sur eux.

À noter une astuce pour sélectionner un élément qui est en recoupement avec d'autres, alors qu'il est derrière: sélectionnez d'abord le premier élément en cliquant dessus puis utilisez la touche ALT lorsque vous cliquez avec le bouton gauche, ce sera l'élément suivant qui sera sélectionné.

Attention: cette fonction modifie les numéros des objets.

Raccourci clavier : AMIGA-M ou M

Appel ARexx: MENU("Passer~devant")      Attention: espace "solide"  
(ALT-ESPACE) entre Passer et devant.

## 1.90 Menu Édition/Passer derrière

Ce menu permet de faire passer un élément en queue de liste. Il est alors choisi en dernier s'il est en recoupement avec d'autres quand l'utilisateur clique sur eux.

Ce menu est bien utile, notamment quand vous cherchez à sélectionner un objet situé en arrière-plan: cliquez dessus et tant que ça n'est pas celui que vous désirez qui se trouve sélectionné, sélectionnez ce menu afin de placer les objets indésirables en arrière-plan.

Attention: cette fonction modifie les numéros des objets.

Raccourci clavier : AMIGA-D ou D

Appel ARexx: MENU("Passer~derrière")      Attention: espace "solide"  
(ALT-ESPACE) entre Passer et derrière.

## 1.91 Menu Édition/Doubler taille

Ce menu permet de doubler la taille des éléments sélectionnés. Cette fonction s'applique aussi à l'élément présent sous le curseur lorsque celui-ci est en mode "placement composant". Elle s'applique également sur un clip qui est en cours de positionnement (voir menu {"Édition/Coller" LINK "Paste~from~clip"}).

La fonction ne prend effet que si l'agrandissement est possible, c'est à dire sans que les éléments ne sortent de la fenêtre.

Raccourci clavier : AMIGA-+ ou +

Voir aussi : fonction SETSCALE.

---

Appel ARexx: MENU("Double~size")      Attention: espace "solide"  
(ALT-ESPACE) entre Double et size.

## 1.92 Menu Édition/Diviser taille

Ce menu permet de diminuer la taille des éléments sélectionnés.  
La taille des éléments doit être un multiple de deux (échelles 2, 4, 8...)  
Cette fonction s'applique aussi à l'élément présent sous le curseur lorsque celui-ci est en mode "placement composant".  
Elle s'applique également sur un clip qui est en cours de positionnement (voir menu {"Édition/Coller" LINK "Paste~from~clip"}).

Raccourci clavier : AMIGA-- ou -

Voir aussi : fonction SETSCALE.

Appel ARexx: MENU("Divide~size")      Attention: espace "solide"  
(ALT-ESPACE) entre Divide et size.

## 1.93 Menu Édition/Étirer

Ce menu permet de changer la taille des éléments sélectionnés en laissant l'utilisateur choisir le rapport d'agrandissement.  
Si le rapport spécifié est supérieur à 100, la taille des objets est augmentée, si elle est inférieure, la taille diminue.

Raccourci clavier : AMIGA-# ou #

Voir aussi : fonction SETSCALE.

Appel ARexx: MENU("Scale")

## 1.94 Menu Édition/Grouper

Ce menu permet d'associer les éléments sélectionnés, en un groupe, leur sélection se fait alors en cliquant sur un seul d'entre eux, cependant leur édition individuelle reste toujours possible, s'il y a lieu, en cliquant deux fois dessus.

Il est impossible d'annuler la sélection d'un seul élément d'un groupe : la sélection d'un de ses éléments ou son annulation est transmise automatiquement aux autres éléments du groupe (vous devez éditer l'un des éléments de ce groupe pour le détacher, en cliquant deux fois sur cet objet, puis cocher la cellule "Détacher cet objet").

Raccourci clavier : AMIGA-{ ou {

Appel ARexx : MENU("Créer~groupe")      Attention: espace "solide"

(ALT-ESPACE) entre Créer et groupe.

## 1.95 Menu Édition/Séparer

Ce menu permet de supprimer les liens définissant un groupe sélectionné, les éléments retrouvent alors leur indépendance. Les éléments sont désélectionnés pour marquer la réussite de la fonction. Ceux qui n'appartiennent pas à un groupe restent sélectionnés.

Pour annuler le lien unissant un seul objet à un groupe, sans détacher les autres, il faut éditer celui-ci en cliquant dessus deux fois à l'aide du bouton gauche de la souris, puis cocher la cellule "Détacher cet objet".

Raccourci clavier : AMIGA-} ou }

Appel ARExx: MENU("Dissocier~groupe")    Attention: espace "solide"  
(ALT-ESPACE) entre Dissocier et groupe.

## 1.96 Menu Édition/Sauver clip

Ce menu permet de sauvegarder le bloc sélectionné dans un fichier (choisi à l'aide d'une requête).

Raccourci clavier : AMIGA-W ou W

Voir aussi: fonctions SAVECLIP, LOADCLIP.

Appel ARExx: MENU("Sauver~clip")    Attention: espace "solide"  
(ALT-ESPACE) entre Sauver et clip.

## 1.97 Menu Édition/Charger clip

Ce menu permet de lire un bloc précédemment sauvé dans un fichier (choisi à l'aide d'une requête). Si la lecture réussit le bloc de données est affiché sous le curseur et se déplace avec lui, utilisez un clic du bouton gauche pour le fixer à l'emplacement souhaité ou bien cliquez avec le bouton droit pour annuler l'opération.

Le point d'accrochage d'un clip est le point le plus haut à gauche, aussi essayez de créer des clips possédant ce point situé sur la grille afin de pouvoir les placer facilement par la suite.

Raccourci clavier : AMIGA-L ou L

Voir aussi: fonctions LOADCLIP, SAVECLIP.

---

Appel ARexx: MENU("Charger~clip")                      Attention: espace "solide"  
(ALT-ESPACE) entre Charger et clip.

## 1.98 Menu Édition/Multisélection

Quand ce menu est marqué, vous pouvez sélectionner autant d'éléments que vous le désirez, en cliquant successivement sur chacun d'eux, sans avoir à utiliser une touche SHIFT. Quand il n'est pas marqué, vous devez utiliser une touche SHIFT pour sélectionner plusieurs éléments, sinon chaque nouvelle sélection annule la précédente.

Il suffit de sélectionner ce menu pour changer son état.

Raccourci clavier : AMIGA-\*

Pas d'appel ARexx possible pour ce menu.

## 1.99 Menu Édition/Restaurer

Ce menu est très pratique pour annuler l'effet de la dernière opération qui a été réalisée. Cependant les effets réalisés à l'aide des fonctions ARexx ne pourront être annulés, sauf si la fonction SAVEALL a été utilisée au début du script.

Raccourci clavier : AMIGA-U ou U

Appel ARexx: MENU("Restaurer")

## 1.100 Menu Scripts

Cette série de menus est dédiée aux scripts ARexx.

Appel~script	Appel d'un script ARexx choisi.
ARexx...	Appel (programmable) de 20 scripts.

## 1.101 Menu Macros

Cette série de menus est dédiée aux macro-commandes. Elle permet l'ajout de menus programmables par l'utilisateur, ceux-ci sont sauves lors d'une opération de sauvegarde des préférences.

Mode direct	Saisie, exécution d'une ligne de commande.
Ajouter menu	Ajout d'un article de menu
Éditer label	Édition d'une entrée de menu
Éditer macro	Édition de la macro associée à un menu
Supprimer menu	Suppression d'un ou de tous les menus

---



## 1.102 Menu Macros/Ajouter menu

Ce menu permet l'ajout de menus définis par l'utilisateur. Pour ajouter une nouvelle entrée à ce menu, il suffit de sélectionner le menu "Ajouter menu": le logiciel ouvre alors une boîte de requête pour demander quelle est la macro commande à associer à ce nouveau menu. Vous pouvez alors saisir n'importe quelle expression utilisant toutes les commandes ARexx définies par AmiCAD, y compris l'appel d'un script ARexx en utilisant la fonction CALL.

Si vous validez votre saisie, le logiciel vous demande ensuite quel est le nom à associer à ce menu, entrez alors un nom définissant le mieux ce que la macro ou le script va faire. La prochaine fois que vous allez sélectionner le menu, ce nom va apparaître en fin de liste, si vous le sélectionnez la commande que vous avez défini précédemment sera exécutée.

ATTENTION: si vous essayez de définir un menu possédant un nom ou une macro identique à ceux d'un des menus déjà définis, votre définition ne sera pas ajoutée à la liste existante.

Pour définir une barre horizontale de séparation, ne donnez rien dans la requête de macro commande (Validez simplement), puis entrez la définition BAR dans la requête demandant le nom du menu.

## 1.103 Menu Macros/Éditer label

Ce menu permet de modifier l'intitulé d'un menu défini par l'utilisateur. Il suffit de sélectionner ce menu puis de cliquer sur le label du menu à modifier dans la fenêtre qui s'ouvre ensuite pour réaliser cette opération.

Spécifiez le label "BAR" pour obtenir une barre de séparation entre des menus successifs.

## 1.104 Menu Utilisateur/Éditer macro

Ce menu permet de modifier la macro-commande associée à un menu défini par l'utilisateur. Il suffit de sélectionner ce menu puis de cliquer sur le label du menu à modifier dans la fenêtre qui s'ouvre ensuite pour réaliser cette opération.

Les macro-commandes peuvent utiliser toutes les fonctions ARexx disponibles. Pour appeler un script ARexx, utilisez la fonction CALL.

## 1.105 Menu Utilisateur/Supprimer menu

---

Ce menu permet de supprimer l'un des menus précédemment définis, s'il en existe. Il suffit de cliquer sur le nom du menu qui vous intéresse, dans la fenêtre ouverte après avoir sélectionné ce menu, pour que l'opération soit effectuée. Cliquez dans le bouton de fermeture de la fenêtre si vous désirez annuler l'opération.

## 1.106 Menu Macros/Mode direct

Ce menu permet la saisie d'une macro-commande, puis son exécution.

Vous pouvez programmer les touches de fonction combinées avec ou sans les touches Shift, Alt et Ctrl pour obtenir des macros préprogrammées (voir touches de fonction).

Il faut noter que ces séquences programmées sont sauvegardées dans le fichier "s:AmiCAD.keys", à l'aide du menu Préférences/Touches/Sauver. Celles-ci sont alors rechargées lors de chaque lancement du programme.

Vous pouvez provoquer l'affichage du résultat de la commande en faisant commencer le texte spécifiant cette commande par le signe égal (=).

Exemple:

=4\*95     affiche le résultat de la multiplication (4 fois 95)  
=FIRSTSEL     affiche le numéro du premier objet sélectionné

Attention: l'opération exécutée sous le contrôle d'une macro ne peut être annulée (menu Édition/Restaurer), à moins d'y utiliser la fonction SAVEALL.

Raccourci clavier : AMIGA-; ou ;

Voir aussi: fonctions EXEC, MAP.

Appel ARexx: MENU("Mode~direct")                      Attention: espace "solide"  
(ALT-ESPACE) entre Mode et direct.

## 1.107 Menu Macros/Appel script

Ce menu permet la saisie d'une commande ARexx (donnez le nom du script, l'extension ".amiCAD" n'est pas obligatoire).  
Si vous validez cette requête avec une chaîne vide la requête de fichier est ouverte afin de choisir le script à exécuter.

Raccourci clavier : AMIGA-, ou ,

Voir aussi : fonction CALL.

Appel ARexx: MENU("Appel~script")                      Attention: espace "solide"  
(ALT-ESPACE) entre Appel et script.

## 1.108 Menu Macros/ARexx...

Ces 20 menus servent à définir 20 commandes pouvant être lancés par la sélection du menu correspondant.

Lors du premier appel du menu, l'utilisateur doit définir sa commande, en donnant le nom du script à exécuter (sans le chemin).

Les appels qui suivront permettent alors l'appel du script défini.

Le nom du script apparaît ensuite dans le menu. Les 20 définitions peuvent être sauvegardées dans le fichier "AmiCAD.prefs", et rechargées automatiquement lors de l'appel du programme.

Ces menus peuvent être redéfinis en appuyant sur l'une des touches SHIFT lors de la sélection du menu.

À noter que l'édition du script défini peut être faite de façon automatique dans une nouvelle fenêtre en appelant le menu tout en maintenant la touche CTRL appuyée (le nom doit naturellement avoir été défini auparavant). Le programme lance alors l'exécution du script ARexx EditScript.AmiCAD en lui passant le nom de ce script comme argument. Ce script (EditScript) doit être adapté à l'éditeur que vous utilisez et à votre installation, il est écrit pour mon éditeur de texte Amitex.

La commande doit comporter le nom du script à exécuter (toujours sans l'extension ".amiCAD"). Le nom ne peut compter que 11 caractères au maximum. Il n'est pas possible de transmettre de paramètres au script.

Les communications avec ARexx se font grâce au port AmiCAD.0

(ou AmiCAD.1, AmiCAD.2...AmiCAD.999 si plusieurs tâches sont exécutées en même temps). Utilisez la commande port=ADDRESS() pour connaître le nom du port, si nécessaire.

Raccourcis clavier : AMIGA-0 à AMIGA-9

Voir aussi : fonction CALL.

Pas d'appel ARexx possible pour ces menus, utilisez la fonction CALL.

## 1.109 Port ARexx

Le programme ouvre normalement un port de communication avec l'interpréteur ARexx lors de son lancement, ce port s'appelle AmiCAD.0. Si le programme est lancé plusieurs fois ou si une nouvelle fenêtre est ouverte, le port est ↵ appelé

AmiCAD.1 pour la seconde tâche, puis AmiCAD.2 pour la suivante, et ainsi de ↵ suite

jusqu'à AmiCAD.999.

## 1.110 Menu Préférences

Cette série de menus permet de sauvegarder certains réglages du programme dans les fichiers AmiCAD.prefs et AmiCAD.keys. Ces fichiers sont sauves dans le répertoire courant.

Ces fichiers sont relus automatiquement à chaque démarrage du

---

programme, afin de repositionner les réglages comme voulu par l'utilisateur. Ces deux fichiers sont au format ASCII et peuvent donc être consultés et même modifiés à l'aide d'un éditeur de texte. Vous pouvez notamment créer des minis-fichiers de configuration ne contenant qu'une partie de la configuration, comme une palette par exemple (voir le fichier AmiCAD.prefs.colors pour l'exemple).

Défilement~auto  
Copie~schéma~sauvé  
Sauver~icône  
Tirer~lignes  
Afficher grille  
Nom complet  
Affichage  
Échelle objets  
Dimensions document  
Palette  
Touches de fonction

### 1.111 Menu Préférences/Échelle objets

Échelle horizontale  
Échelle verticale

### 1.112 Menu Préférences/Défilement auto

Quand ce menu est marqué, si le curseur arrive à une bordure de la fenêtre de travail, le défilement du contenu de la feuille de schéma se fait automatiquement s'il y a lieu.

Il n'y a pas de raccourci clavier pour cette fonction.

Pas d'appel ARexx possible pour ces menus.

### 1.113 Menu Préférences/Copie schéma sauvé

Quand ce menu est marqué, lors de la sauvegarde, l'ancienne version est sauvegardée avec l'extension ".bis" à la fin du nom.

Il n'y a pas de raccourci clavier pour cette fonction.

Voir aussi : fonction SAVECOPY.

Pas d'appel ARexx possible pour ce menu.

---

### 1.114 Menu Préférences/Sauver icône

Si ce menu est marqué une icône est créée lors de la sauvegarde du texte. Le nom du programme figure dans le champ TOOL-TYPES. L'icône utilisée est l'icône "Projet" par défaut, stockée dans le répertoire ENV:.

Il n'y a pas de raccourci clavier pour cette fonction.

Voir aussi : fonction SAVEICON.

Pas d'appel ARexx possible pour ce menu.

### 1.115 Menu Préférences/Tirer lignes

Si ce menu est marqué, lors du déplacement d'un ou plusieurs composants à l'aide la souris, les fils connectés à ces composants seront également déplacés. Attention : cette opération n'a lieu qu'après avoir lâché ces composants à leur emplacement de destination.

Raccourci clavier : AMIGA-F ou F

Appel ARexx : MENU("Tirer")

### 1.116 Menu Préférences/Afficher grille

Si ce menu est marqué, une grille est affichée sur l'écran. Le pas utilisé est celui défini dans le menu Dessin/Pas grille. L'écran est mis à jour après l'appel du menu.

Il n'y a pas de raccourci clavier pour cette fonction.

Pas d'appel ARexx possible pour ce menu.

### 1.117 Menu Préférences/Nom complet

Ce menu permet d'afficher le nom complet du document dans la barre de titre, s'il est marqué. S'il ne l'est pas, le nom est affiché sans le chemin, ceci permet d'obtenir les indications qui suivent, les coordonnées du curseur notamment, quand la fenêtre n'est pas suffisamment large ou quand le nom complet est très long (répertoires et sous répertoires multiples).

## 1.118 Menu Préférences/Affichage

Cet ensemble de menus permet de choisir les caractéristiques de l'affichage, particulièrement l'échelle du zoom. Le coefficient de zoom peut être réglé par l'utilisateur ou par l'un des menus suivants dans une plage de 25% à 1000%.

Un coefficient égal à 100% correspond à une vue normale (échelle 1).

- 0- Largeur écran
- 4- Hauteur écran
- 1- Échelle normale
- 2- Zoom boîte
- 5- Recentrer
- 6- Doubler échelle
- 9- Diviser échelle
- 3- Saisie échelle

Voir aussi : fonction ARexx ZOOM.

## 1.119 Menu Préférences/Échelle horizontale

La valeur associée à ce menu permet de définir l'échelle par défaut des composants qui seront placés sur le schéma.

Il n'y a pas de raccourci clavier pour cette fonction.

Voir aussi : fonctions HSCALE, SETSCALE.

Appel ARexx: MENU("Échelle~horizontale")    Attention: espace "solide"  
(ALT-ESPACE) entre Échelle et horizontale.

## 1.120 Menu Préférences/Échelle verticale

La valeur associée à ce menu permet de définir l'échelle par défaut des composants qui seront placés sur le schéma.

Il n'y a pas de raccourci clavier pour cette fonction.

Voir aussi : fonctions VSCALE, SETSCALE.

Appel ARexx: MENU("Échelle~verticale")    Attention: espace "solide"  
(ALT-ESPACE) entre Échelle et verticale.

## 1.121 Menu Préférences/Dimensions document

---

Ce menu permet de définir la largeur et la hauteur de la fenêtre de travail. La fenêtre est en fait du type SuperBitmap, ce qui permet de bénéficier de défilements très rapides, mais exige une quantité mémoire "chip" importante.

Il n'y a pas de raccourci clavier pour cette fonction.

Voir aussi: fonction DIMSHEET.

Appel ARexx: MENU("Dimensions")

## 1.122 Menu Préférences/Palette

Ce menu est maintenant intégré à la requête déterminant les préférences au démarrage (onglet palette).

Cette fonction permet de choisir les couleurs de l'écran utilisé par AmiCAD. À noter que ces couleurs sont sauveées dans le fichier de configuration. Le programme n'utilise que 16 couleurs.

Les couleurs utilisées par le programme sont respectivement:

- la couleur 6 pour la grille,
- la couleur 7 pour remplir le fond des fenêtres de requêtes,
- la couleur 8 pour les composants,
- la couleur 9 pour les fils, les ellipses, les arcs de cercles et les jonctions, ↔
- la couleur 10 pour les connecteurs d'entrée,
- la couleur 11 pour les connecteurs de sortie,
- la couleur 12 pour les références des composants,
- la couleur 13 pour les valeurs des composants,
- la couleur 14 pour les textes,
- la couleur 15 pour signaler les éléments en cours de sélection.

Si vous préférez que les couleurs soient toutes identiques, comme auparavant, réglez les couleurs 8 à 14 avec les mêmes niveaux RVB.

Les couleurs 0 à 3 sont utilisées par le système pour dessiner les fenêtres, les menus, boutons, etc...

## 1.123 Menu Préférences/Touches de fonction

Ce menu permet de programmer les touches de fonction. Il est apparu dans la version 2.04 du programme. Pour sauver la configuration, utilisez le bouton Sauver (fichier AmiCAD.keys).

À noter que les définitions des autres touches (définies à l'aide de la fonction MAP) sont aussi sauveées dans le fichier AmiCAD.keys.

## 1.124 Touches - Raccourcis clavier

---

Certaines touches donnent accès à des fonctions :

- \* (clavier numérique) : redessine l'écran
- ENTER, RETURN: permet l'ouverture de la requête d'édition d'un objet s'il y en a un qui est sélectionné, équivalent à un clic sur le bouton gauche de la souris sinon (mode placement)
- BACKSPACE: annulation de la dernière modification
- Flèches curseur: déplacement des éléments sélectionnés (peut être combiné avec la touche SHIFT, pour un déplacement plus rapide)
- En combinaison avec les touches ALT et CTRL permet de modifier le placement d'une ligne (début ou fin).
- SPACE: passe en mode placement de fil.
- DEL: suppression des objets sélectionnés
- HELP: ouverture du guide

### 1.125 Menu Préférences/Affichage/0- Largeur écran

La sélection de ce menu entraîne une modification de la fenêtre courante, celle-ci est alors dimensionnée de façon à ce que tout la pleine largeur du document soit visible en occupant toute la largeur de l'écran.

### 1.126 Menu Préférences/Affichage/4- Hauteur écran

La sélection de ce menu entraîne une modification de la fenêtre courante, celle-ci est alors dimensionnée de façon à ce que tout la pleine hauteur du document soit visible en occupant toute la hauteur de l'écran.

### 1.127 Menu Préférences/Affichage/0- Échelle 1

La sélection de ce menu entraîne un affichage à l'échelle 1 (100%).

### 1.128 Menu Préférences/Affichage/2- Zoom boîte

La sélection de ce menu permet de sélectionner la zone devant être représentée à l'écran, à l'aide de la souris. Il suffit alors de cliquer une première fois dans un angle de cette zone à l'aide du bouton gauche de la souris, puis dans le second après l'y avoir déplacée. Le coefficient de zoom sera alors automatiquement ajusté au mieux.

L'appui sur une touche du clavier ou sur le bouton droit de la souris entraîne l'arrêt de cette fonction.

---



### 1.129 Menu Préférences/Affichage/5- Recentrer

La sélection de ce menu permet de recentrer l'affichage sur le point désiré. Cliquez sur ce point à l'aide du bouton gauche de la souris après avoir sélectionné ce menu.

### 1.130 Menu Préférences/Affichage/6- Doubler échelle

La sélection de ce menu entraîne le doublement du coefficient de zoom, jusqu'à concurrence de 1000%.

### 1.131 Menu Préférences/Affichage/9- Diviser échelle

La sélection de ce menu entraîne la division par 2 du coefficient de zomm, jusqu'à concurrence de 25%.

### 1.132 Menu Préférences/Affichage/3- Saisie échelle

La sélection de ce menu entraîne l'ouverture d'une boîte dans laquelle l'utilisateur est invité à saisir le coefficient de zoom qu'il souhaite utiliser.

### 1.133 Format des fichiers de configuration

Les fichiers AmiCAD.prefs et AmiCAD.keys sont des fichiers au format ASCII, ils peuvent ainsi être consultés et (aussi modifiés) à l'aide d'un simple éditeur de texte. Leur format répond cependant à quelques règles élémentaires.

Chacun de ces deux fichiers commence par un en-tête qui ne doit pas être changé, il comprend notamment le numéro de version de ces fichiers, AmiCAD ne reconnaît que le numéro de version en vigueur pour une version donnée: la version 1.2 de AmiCAD attend l'en-tête "AmiCADPrefs 1.1" pour le fichier AmiCAD.prefs et l'en-tête "AmiCADKeys 1.0" pour le fichier AmiCAD.keys.

Le fichier AmiCAD.prefs comprend ensuite plusieurs paragraphes, ces paragraphes peuvent être supprimés pour créer une configuration particulière, comprenant par exemple uniquement une palette de couleurs (fichier AmiCAD.palette par exemple). Chacun de ces paragraphes débute par un texte placé entre deux crochets ([Menus\_ARexx], [Screen], [Palette] et [Mode]), ces textes sont suivis de plusieurs lignes comprenant chacune des informations différentes. Tout ou partie de ces lignes peuvent être supprimées.

De la même façon le fichier AmiCAD.keys comprend deux paragraphes

---

[FunctionKeys] et [Macros]. Vous pouvez ajouter ou supprimer des lignes dans chacun de ces paragraphes, à votre convenance.

Attention: les mots situés à gauche des signes = doivent être écrits tels quels (pas de changement majuscules/minuscules).

## 1.134 Écriture d'une ligne de commande(s)

Les macro-commandes peuvent être exécutées en mode local ou bien par l'intermédiaire d'un script ARexx. Vous pouvez vous reporter aux scripts donnés en exemple, dans le répertoire ARexx. Ceux-ci doivent posséder l'extension ".amiCAD" à leur nom.

Chacune de ces macro-commandes peut appeler l'une des fonctions ARexx, ou même plusieurs. Certaines d'entre elles nécessitent un ou plusieurs arguments, enfin la plupart d'entre elles retournent un résultat. Le programme permet aussi de manipuler des variables (des types numérique ou chaîne de caractères). De plus vous pouvez définir vos propres fonctions. L'appel d'une fonction se fait en donnant son nom, suivi d'un ou plusieurs arguments, entourés de parenthèses, éventuellement séparés par des virgules.

Les opérateurs mathématiques classiques sont bien sûr disponibles :

opérateur	signe	priorité
élévation à une puissance :	^	10
division :	/	9
multiplication :	*	9
modulo :	%	9
addition :	+	8
soustraction :	-	8
ET logique :	&	6
OU exclusif :	?	5
OU logique :		4
affectation :	=	2

Ce dernier opérateur (=) peut être utilisé pour affecter les variables.

Ex : A=2, affecte la valeur 2 à la variable A.

Il est à noter que l'écriture A=B=3 n'est pas admise, elle entraînera un message d'erreur "Affectation impossible". Utilisez plutôt l'écriture A=3;B=3

Autres opérateurs :

décalage à gauche : <<

décalage à droite : >>

Ces deux opérateurs ont une priorité égale à 7.

test supérieur : >      test supérieur ou égal : >=

test inférieur : <      test inférieur ou égal : <=

test si différent : <>      test égalité : ==

Les tests renvoient la valeur 1 s'ils sont vérifiés, 0 dans les autres cas. Leur priorité est égale à 3.

Il existe un opérateur particulier qui en fait n'en est pas un, c'est le signe :, celui-ci permet en fait de séparer deux formules (ou davantage) afin de pouvoir saisir plusieurs affectations de variables dans une même fonction. Ainsi vous pouvez saisir A=2;B=3;C=0 dans une seule commande au lieu d'en utiliser trois. Le résultat renvoyé est alors celui de la

dernière opération réalisée (0 dans cet exemple). Cette possibilité peut également présenter un intérêt pour une fonction FOR (voir plus loin), afin de réaliser de multiples initialisations au début d'une boucle.

Enfin il est à noter que vous pouvez placer un commentaire dans une macro comportant une expression mathématique en utilisant l'apostrophe vue plus haut.

Exemple:

```
DELETE(FIRSTSEL) ' Effacement du premier élément sélectionné
```

Enfin, vous pouvez placer plusieurs appels de fonctions sur une même ligne, en les séparant par deux points.

Exemple:

```
SETSCALE(FIRSTSEL,100,200):SETSCALE(NEXTSEL(FIRSTSEL),100,200)
```

## 1.135 Les variables

Les variables manipulées par AmiCAD peuvent être de deux types: numérique ou chaîne de caractères. Le type est choisi lors de l'affectation, il ne peut ensuite être changé, à moins de réinitialiser cette variable (voir fonction INIT).

Pour affecter une valeur à une variable il suffit de faire suivre son nom du signe = et de la valeur à lui affecter.

Exemples:

```
A = 1
```

```
B2 = "chaîne de texte"
```

```
A2 = "Première "+B2      donne "Première chaîne de texte"
```

Les chaînes de caractères doivent être encadrées par des guillemets s'il s'agit de constantes. Les valeurs numériques sont limitées aux entiers (longs), utilisez ARexx pour manipuler des nombres réels.

Les noms de variables peuvent comprendre de 1 à 21 caractères, ils doivent débuter par une lettre, les caractères suivants pouvant être des lettres (accentuées ou non), des nombres ou le caractère \_.

Exemples de noms valides:

```
ESSAI
```

```
TYPE_DONNÉE
```

```
LIGNE1
```

À noter que le programme convertit normalement les noms de fonctions et de variables en majuscules lors de la saisie, aussi n'y aura-t-il pas de différence entre deux noms comme variable et VARIABLE (ou même Variable), cependant dans un script ARexx, utilisez de préférence les majuscules car les chaînes issues de l'interpréteur ne sont pas forcément converties, et le programme différenciera alors les noms écrits en minuscules et/ou en majuscules.

Attention: les variables ne sont pas utilisables directement avec ARexx, elles sont internes au programme AmiCAD et l'interpréteur ARexx possède ses propres variables. Il est cependant possible de renvoyer une valeur de variable à l'interpréteur simplement en donnant son nom:

Exemple:

```
'NOM_VARIABLE'; v=result
```

Le contenu de la variable NOM\_VARIABLE est ainsi transféré dans la variable ARexx v.

Il est souvent avantageux d'utiliser des variables internes à AmiCAD car leur traitement est plus rapide, d'autre part cela peut éviter l'envoi de certains messages entre l'interpréteur et le programme, ce qui entraîne un gain de vitesse énorme (voir scripts Zoom et Unzoom).

## 1.136 Les variables numériques

Les nombres manipulés par AmiCAD sont limités aux nombres entiers. Ceux ci sont des entiers longs signés, sur 32 bits. La valeur maximale est doc de  $2^{31}-1$  (2147483647) et la valeur minimale  $-2^{31}$  (-2147483648). Utilisez les capacités de l'interpréteur ARexx pour manipuler des nombres en virgule flottante.

## 1.137 Les chaînes de caractères

Les chaînes de caractères comprennent un nombre quelconque de caractères (théoriquement limité uniquement par la capacité mémoire de votre micro-ordinateur), en commençant par les chaînes nulles qui ne comprennent aucun caractère. Pour définir une chaîne de caractères, vous devez encadrer son contenu par deux guillemets. Si cette chaîne comprend elle-même un ou plusieurs guillemets, ceux-ci doivent être doublés.

Exemples:

```
"Ceci est une chaîne de caractères"
```

```
""
```

```
"C'est un guillemet "" !"
```

## 1.138 Structure des scripts ARexx

Les scripts ARexx sont des fichiers ASCII, se conformant à un format permettant leur interprétation par ce logiciel. Ils peuvent inclure toutes les fonctions spécifiques à ARexx et à ses bibliothèques (reportez-vous à leurs documentations pour plus de précisions), ainsi que toutes les commandes supportées par le programme AmiCAD (une centaine, plus les fonctions éventuellement définies par vos soins).

Emplacement

Ces scripts doivent se situer soit dans le répertoire AmiCAD/ARexx. Il est conseillé de les nommer avec l'extension .AmiCAD, mais ce n'est pas obligatoire.

Format

Ces scripts doivent toujours commencer par une remarque

---

incluse entre les caractères /\* et \*/, comme en langage C.

Les commandes AmiCAD doivent impérativement figurer en majuscules. Ce sont les mêmes que celles qui sont utilisées en mode local, cependant leur analyse par ARexx nécessite quelques précautions, ainsi ces commandes doivent impérativement être encadrées par des apostrophes (') ou des guillemets, afin de les différencier des fonctions internes à ARexx.

#### Exemple

La structure suivante est recommandée pour ces scripts:

```
/* Ce fichier vous donne un exemple de structure possible
pour un fichier de macros appelé par AmiCAD, par le menu Macros */

options results      /* indispensable pour récupérer le résultat des macros */

signal on error      /* pour l'interception des erreurs */
signal on syntax

/* votre programme doit être situé dans cette zone */

exit

/* Traitement des erreurs, interruption du programme */
syntax:
erreur=RC
'MESSAGE("Erreur de syntaxe"+CHR(10)+"en ligne 'SIGL'"+CHR(10)+"'error text(erreur) ←
'")'
exit

error:
'MESSAGE("Erreur en ligne 'SIGL'")'
exit
```

Cet exemple existe déjà prêt dans le répertoire ARexx, sous le nom squelette.AmiCAD.

## 1.139 Liste des fonctions ARexx

ABS	valeur absolue d'un nombre
ARC	trace un arc de cercle
ARRAYDIM	lecture du nombre d'éléments d'un tableau
ASC	code ASCII d'un caractère
ASKNUM	saisie d'un nombre
ASKTEXT	saisie d'une chaîne de caractères
BLINK	clignotement d'un objet
BOX	trace un rectangle
CALL	appel d'un script ARexx
CHR	renvoie le caractère possédant le code ASCII spécifié
CLIPPATH	choix du tiroir contenant les clips
CLIPUNIT	choix de la zone mémoire utilisée pour les opérations couper/coller
CLOSE	ferme la fenêtre texte spécifiée
COL	numéro de colonne où est situé un objet

---

CONVERT	choix symbole alterné pour composant
COORDS	lecture des coordonnées d'un objet
COPY	copie d'un ou plusieurs objets en mémoire
CURSMODE	place le curseur en mode placement objet
DATE	renvoie la date actuelle
DEF	définition d'une nouvelle fonction
DELARRAY	suppression d'un tableau de données
DELETE	suppression d'un ou plusieurs objets
DEVPINS	nombre de connexions d'un composant
DIMARRAY	dimensionnement d'un tableau de données
DIMSHEET	dimensionne la fenêtre
DRAW	tracé d'une ligne
DRAWMODE	sélection du type de tracé
EDIT	appel de la requête d'édition
ELLIPSE	tracé d'une ellipse ou d'un cercle
ENDCOL	numéro de colonne de fin d'une ligne
ENDLINE	numéro de ligne de fin d'une ligne
EXEC	interprétation d'une chaîne de caractères
FILENAME	nom du fichier complet
FILEPART	nom du fichier
FINDLAB	recherche d'un label
FINDLINE	recherche du passage d'une ligne
FINDOBJ	test présence objet
FINDPART	recherche d'un composant
FINDREF	recherche d'un composant par sa référence
FINDVAL	recherche d'un composant par sa valeur
FIRSTSEL	renvoie le numéro du premier objet sélectionné
FONTNAME	renvoie le nom de la fonte de caractères utilisée
FONTSIZE	renvoie la taille de la fonte de caractères utilisée
FOR	traitement d'une boucle
GETARRAY	retourne la valeur d'un élément de tableau
GETCOLOR	lecture des niveaux de couleur
GETDEVS	lecture du nombre de portes d'un circuit
GETLABEL	lecture du label associé à un fil
GETLINE	définition d'une ligne à l'aide de la souris
GETNET	lecture de la ligne associée à un label
GETPOINT	choix d'un point à l'aide de la souris
GETPOS	lecture du mode de placement d'un composant
GETREF	lecture de la référence d'un composant
GETVAL	lecture de la valeur ou du type d'un composant
GETZONE	définition d'une zone à l'aide de la souris
GROUP	crée un nouveau groupe
HEIGHT	renvoie la largeur d'un objet
HELP	affichage d'une aide AmigaGuide
HSCALE	renvoie la valeur de l'échelle horizontale d'un objet
IF	test
INIT	initialisation de variables
INPUT	placement d'un connecteur d'entrée
JUNCTION	placement d'une jonction
LANGUAGE	renvoie le nom de la "locale"
LEN	longueur d'une chaîne de caractères
LIBRARY	renvoie le nom de la bibliothèque associée à un composant
LIBSPATH	choix du tiroir contenant les bibliothèques de symboles
LINE	numéro de ligne où est situé un objet
LINKLAB	affectation d'un label à un fil
LINKREF	affectation d'une référence à un composant
LINKVAL	affectation d'une valeur à un composant

---

---

LOAD	chargement d'un schéma
LOADCLIP	chargement d'un clip
LOADLIB	chargement d'une bibliothèque de symboles
LOCK	verrouillage saisies utilisateur
MAP	programmation d'une macro clavier
MARK	marquage d'un ou plusieurs objets
MARKZONE	marquage des éléments compris dans la zone spécifiée
MENU	exécution de la fonction associée à un menu
MESSAGE	affichage d'un message
MESURE	renvoie une dimension de la fenêtre
MODIF	test modification texte
MOVE	déplacement d'un objet
NBSHEET	nombre de schémas présents en mémoire
NBSHEETS	nombre de fenêtres, noms des ports ARexx
NEWSHEET	ouverture d'une nouvelle fenêtre
NEXTSEL	numéro du prochain élément sélectionné
OBJECTS	renvoie le nombre d'éléments d'un schéma
OPEN	ouverture de schéma(s)
OUTPUT	placement d'un connecteur de sortie
PARTNAME	lecture du nom d'un composant
PASTE	collage du contenu d'un tampon mémoire
PENWIDTH	établit la largeur du trait
PICKOBJ	choix d'un objet à l'aide de la souris
PINCOL	emplacement (colonne) d'une broche
PINLINE	emplacement (ligne) d'une broche
PINNUM	numéro de broche
PRINT	impression du schéma
PUTPART	placement d'un composant
READCONV	lecture du type de symbole
READDEF	lecture de la définition d'une fonction
READDEV	lecture du numéro de circuit d'un composant
READMAP	lecture d'une macro clavier
READTEXT	lecture d'un texte associé à un objet
REMLIB	suppression d'une bibliothèque
REQFILE	choix d'un fichier
REQLIST	choix d'une chaîne parmi n
REQSHEET	choix d'un schéma
REQUEST	affichage d'un message, choix OUI/NON
RESET	réinitialisation de variables
REXXPORT	nom du port ARexx
ROTATE	rotation d'un objet
SAVEIFF	sauvegarde au format IFF
SAVE	sauvegarde d'un schéma
SAVEALL	sauvegarde du contenu du document en mémoire
SAVECLIP	sauvegarde d'un clip
SAVECOPY	test/choix sauvegarde copie fichier
SAVEICON	test/choix sauvegarde avec création icône
SCREEN	choix du mode écran
SCRMODE	renvoie le mode écran
SECURITY	détermination d'un nombre de boucles maximum avant débordement
SELECT	choix d'une option parmi plusieurs
SETARRAY	affectation d'une valeur à un élément de tableau
SETCOLOR	choix couleur écran
SETDEV	choix du circuit ou d'une porte de composant
SETFILL	sélection du mode remplissage
SETGRID	choix du pas de la grille
SETLABEL	fixe un label à un fil

---

SETORTHO détermine la façon d'effectuer les tracés  
SETPINS affichage numéros de broches  
SETREF fixe la référence d'un objet  
SETSCALE détermination des échelles horizontale et verticale  
SETTEXT fixe le texte d'un objet  
SETVAL fixe la valeur ou le type d'un objet  
SGN test du signe d'un nombre  
SHEIGHT renvoie la hauteur de l'écran  
SNAPGRID fixe l'état du menu aligner sur grille  
STOBACK passage de l'écran en arrière-plan  
STOFRONT passage de l'écran an avant-plan  
STR conversion d'un nombre en chaîne de caractères (décimal)  
SWIDTH renvoie la largeur de l'écran  
SYMMETRY symétrie d'un objet  
TEST test si objet sélectionné  
TIME renvoie l'heure courante  
TITLE choix du titre affiché dans la fenêtre  
TOOLBAR chargement d'une barre d'outils  
TRIANGLE trace un triangle  
TYPE renvoie le type d'un objet  
TXHEIGHT renvoie la hauteur occupée par un texte  
TXWIDTH renvoie la largeur occupée par un texte  
UNLINK suppression des liens d'un composant  
UNLOCK annule le verrouillage des actions utilisateur  
UNMAP annulation d'une macro clavier  
UNMARK annulation du marquage d'un objet  
VAL conversion d'une chaîne de caractères en nombre  
VERSION numéro de version du programme  
VSCALE renvoie la valeur de l'échelle verticale d'un objet  
WHEIGHT renvoie la hauteur totale du plan de travail  
WHILE boucle tant\_que...  
WIDTH renvoie la largeur d'un objet  
WINDOW dimensionne la fenêtre schéma  
WRITE placement d'un texte  
WTOBACK renvoie la fenêtre active en arrière-plan  
WTOFRONT renvoie la fenêtre active en avant-plan  
WWIDTH renvoie la largeur totale du plan de travail  
ZOOM gestion de l'affichage  
Index thématique

## 1.140 Classement thématique des fonctions ARexx

Traitement des chaînes de caractères  
Placement de nouveaux objets sur le schéma  
Édition, modification des objets existants  
Fonctions portant sur des blocs d'objets  
Fonctions mathématiques  
Fonctions interactives  
Gestion des fenêtres  
Gestion des préférences  
Fonctions diverses



### 1.141 Fonctions ARexx de traitement des chaînes de caractères

ASC code ASCII d'un caractère  
CHR renvoie le caractère possédant le code ASCII spécifié  
LEN longueur d'une chaîne de caractères  
STR conversion d'un nombre en chaîne de caractères (décimal)  
VAL conversion d'une chaîne de caractères en nombre

### 1.142 Fonctions ARexx permettant le placement de nouveaux objets

ARC trace un arc de cercle  
BOX trace un rectangle  
CURSMODE place le curseur en mode placement objet  
DELETE suppression d'un ou plusieurs objets  
DRAW tracé d'une ligne  
DRAWMODE sélection du type de tracé  
ELLIPSE tracé d'une ellipse ou d'un cercle  
GROUP crée un nouveau groupe  
INPUT placement d'un connecteur d'entrée  
JUNCTION placement d'une jonction  
OUTPUT placement d'un connecteur de sortie  
PUTPART placement d'un composant  
SETFILL sélection du mode remplissage  
SETORTHO détermine la façon d'effectuer les tracés  
SNAPGRID fixe l'état du menu aligner sur grille  
TRIANGLE trace un triangle  
UNGROUP dissocie un groupe  
WRITE placement d'un texte

### 1.143 Fonctions ARexx permettant de gérer les blocs d'objets

CLIPUNIT choix de la zone mémoire utilisée pour les opérations couper/coller  
COL numéro de colonne où est situé le bloc courant (COL(0))  
COPY copie d'un ou plusieurs objets en mémoire  
FIRSTSEL renvoie le numéro du premier objet sélectionné  
HEIGHT largeur du bloc courant (HEIGHT(0))  
LINE numéro de ligne où est situé le bloc courant (LINE(0))  
LOADCLIP chargement d'un clip  
MARK marquage d'un ou plusieurs objets  
MARKZONE marquage des éléments compris dans la zone spécifiée  
NEXTSEL numéro du prochain élément sélectionné  
PASTE collage du contenu d'un tampon mémoire  
SAVEALL sauvegarde du contenu du document en mémoire  
SAVECLIP sauvegarde d'un clip  
TEST test si objet sélectionné  
UNMARK annulation du marquage d'un objet  
WIDTH renvoie la largeur du bloc courant (WIDTH(0))

### 1.144 Fonctions ARexx permettant de gérer les objets

---

---

BLINK	clignotement d'un objet
CLIPPATH	choix du tiroir contenant les clips
COL	numéro de colonne où est situé un objet
CONVERT	choix symbole alterné pour composant
COORDS	lecture des coordonnées d'un objet
DELETE	suppression d'un ou plusieurs objets
DEVPINS	nombre de connexions d'un composant
EDIT	appel de la requête d'édition
ENDCOL	numéro de colonne de fin d'une ligne
ENDLINE	numéro de ligne de fin d'une ligne
FINDLAB	recherche d'un label
FINDLINE	recherche du passage d'une ligne
FINDOBJ	test présence objet
FINDPART	recherche d'un composant
FINDREF	recherche d'un composant par sa référence
FINDVAL	recherche d'un composant par sa valeur
GETDEVS	lecture du nombre de portes d'un circuit
GETLABEL	lecture du label associé à un fil
GETNET	lecture de la ligne associée à un label
GETPOS	lecture du mode de placement d'un composant
GETREF	lecture de la référence d'un composant
GETVAL	lecture de la valeur ou du type d'un composant
GROUP	crée un nouveau groupe
HEIGHT	largeur d'un objet
HSCALE	renvoie la valeur de l'échelle horizontale d'un objet
LIBRARY	renvoie le nom de la bibliothèque associée à un composant
LIBSPATH	choix du tiroir contenant les bibliothèques de symboles
LINE	numéro de ligne où est situé un objet
LINKLAB	affectation d'un label à un fil
LINKREF	affectation d'une référence à un composant
LINKVAL	affectation d'une valeur à un composant
LOADLIB	chargement d'une bibliothèque de symboles
MOVE	déplacement d'un objet
OBJECTS	renvoie le nombre d'éléments d'un schéma
PARTNAME	lecture du nom d'un composant
PENWIDTH	établit la largeur du trait
PINCOL	emplacement (colonne) d'une broche
PINLINE	emplacement (ligne) d'une broche
PINNUM	numéro de broche
READCONV	lecture du type de symbole
READDEV	lecture du numéro de circuit d'un composant
READTEXT	lecture d'un texte associé à un objet
REMLIB	suppression d'une bibliothèque
ROTATE	rotation d'un objet
SETDEV	choix du circuit ou d'une porte de composant
SETLABEL	fixe un label à un fil
SETPINS	affichage numéros de broches
SETREF	fixe la référence d'un objet
SETSCALE	détermination des échelles horizontale et verticale
SETTEXT	fixe le texte d'un objet
SETVAL	fixe la valeur ou le type d'un objet
SYMMETRY	symétrie d'un objet
TYPE	renvoie le type d'un objet
TXHEIGHT	renvoie la hauteur occupée par un texte
TXWIDTH	renvoie la largeur occupée par un texte
UNGROUP	dissocie un groupe

---

UNLINK suppression des liens d'un composant  
VSCALE renvoie la valeur de l'échelle verticale d'un objet  
WIDTH renvoie la largeur d'un objet

### 1.145 Fonctions ARexx de gestion des préférences

FONTNAME nom de la fonte de caractères utilisée  
FONTSIZE taille de la fonte de caractères utilisée  
GETCOLOR lecture des niveaux de couleur  
LANGUAGE renvoie le nom de la "locale"  
SAVECOPY test/choix sauvegarde copie fichier  
SAVEICON test/choix sauvegarde avec création icône  
SCREEN choix du mode écran  
SCRMODE renvoie le mode écran  
SETCOLOR choix couleur écran  
SETGRID choix du pas de la grille  
SHEIGHT renvoie la hauteur de l'écran  
SNAPGRID fixe l'état du menu aligner sur grille  
SWIDTH renvoie la largeur de l'écran  
TOOLBAR chargement d'une barre d'outils

### 1.146 Fonctions ARexx de calcul

ABS valeur absolue d'un nombre  
FOR traitement d'une boucle  
IF test  
INIT initialisation de variables  
RESET réinitialisation de variables  
SECURITY nombre de boucles maximum avant débordement  
SGN test du signe d'un nombre  
STR conversion d'un nombre en chaîne de caractères (décimal)  
VAL conversion d'une chaîne de caractères en nombre  
WHILE boucle tant\_que...

### 1.147 Fonctions ARexx permettant le dialogue avec l'utilisateur

ASKNUM saisie d'un nombre  
ASKTEXT saisie d'une chaîne de caractères  
CURSMODE place le curseur en mode placement objet  
GETLINE définition d'une ligne à l'aide de la souris  
GETPOINT choix d'un point à l'aide de la souris  
GETZONE définition d'une zone à l'aide de la souris  
LOCK verrouillage saisies utilisateur  
MESSAGE affichage d'un message  
PICKOBJ choix d'un objet à l'aide de la souris  
REQFILE choix d'un fichier  
REQLIST choix d'une chaîne parmi n  
REQUEST affichage d'un message, choix OUI/NON  
SELECT choix d'une option parmi plusieurs  
UNLOCK annule le verrouillage des actions utilisateur

---

## 1.148 Fonctions ARexx diverses

ARRAYDIM	lecture du nombre d'éléments d'un tableau
CALL	appel d'un script ARexx
DATE	renvoie la date actuelle
DEF	définition d'une nouvelle fonction
EXEC	interprétation d'une chaîne de caractères
DELARRAY	suppression d'un tableau de données
DIMARRAY	dimensionnement d'un tableau de données
FINDLAB	recherche d'un label
FINDLINE	recherche du passage d'une ligne
FINDOBJ	test présence objet
FINDPART	recherche d'un composant
GETARRAY	retourne la valeur d'un élément de tableau
HELP	affichage d'une aide AmigaGuide
MAP	programmation d'une macro clavier
MENU	exécution de la fonction associée à un menu
READDEF	lecture de la définition d'une fonction
READMAP	lecture d'une macro clavier
SETARRAY	affectation d'une valeur à un élément de tableau
TIME	renvoie l'heure courante
UNMAP	annulation d'une macro clavier
VERSION	numéro de version du programme

## 1.149 Fonctions ARexx de gestion des fenêtres

CLOSE	ferme la fenêtre texte spécifiée
DIMSHEET	dimensionne la fenêtre
FILENAME	nom du fichier complet
FILEPART	nom du fichier
FONTNAME	nom de la fonte de caractères utilisée
FONTSIZE	taille de la fonte de caractères utilisée
LOAD	chargement d'un schéma
MESURE	renvoie une dimension de la fenêtre
MODIF	test modification texte
NBSHEET	nombre de schémas présents en mémoire
NBSHEETS	nombre de fenêtres, noms des ports ARexx
NEWSHEET	ouverture d'une nouvelle fenêtre
OPEN	ouverture de schéma(s)
PRINT	impression du schéma
REQSHEET	choix d'un schéma
REXXPORT	nom du port ARexx
SAVEIFF	sauvegarde au format IFF
SAVE	sauvegarde d'un schéma
SAVEALL	sauvegarde du contenu du document en mémoire
SETCOLOR	choix couleur écran
SETGRID	choix du pas de la grille
STOBACK	passage de l'écran en arrière-plan
STOFRONT	passage de l'écran en avant-plan
TITLE	choix du titre affiché dans la fenêtre
WHEIGHT	renvoie la hauteur totale de la fenêtre
WINDOW	dimensionne la fenêtre schéma
WTOBACK	renvoie la fenêtre active en arrière-plan
WTOFRONT	renvoie la fenêtre active en avant-plan

---

WWIDTH    renvoie la largeur totale du plan de travail  
ZOOM       gestion de l'affichage

## 1.150 Fonction ARexx DEF

Vous pouvez définir un nombre quelconque de fonctions utilisant des fonctions internes ainsi que n'importe quel opérateur. Une de ces fonctions peut même faire appel à une autre fonction précédemment définie. Les arguments peuvent être d'un type quelconque, il doivent simplement correspondre aux types attendus par les fonctions qui seront appelées ou être compatibles avec les opérateurs utilisés.

Il ne peut y avoir plus d'une déclaration dans une même ligne.

La forme de ces définitions est de la forme suivante:

```
DEF nom_fonction(argument,...) = définition des opérations
```

Le mot clé DEF doit impérativement commencer la définition, sans aucun espace préalable.

Le nom des fonctions peut comprendre de 1 à 13 caractères alphanumériques, y compris les lettres accentuées et le soulignement (\_). Le premier caractère doit cependant toujours être une lettre. Les fonctions internes ne peuvent être redéfinies. Le nombre des arguments est limité à 15 au maximum. Ce nombre est fixe pour une définition donnée (vous ne pouvez pas déclarer de fonction possédant un nombre variable d'arguments). Il doit y avoir au moins un argument dans la définition, mais celui-ci peut ne pas être utilisé. Les arguments doivent bien entendu posséder des noms distincts.

Exemples :

```
DEF EXPAND(OBJET) = SETSCALE(OBJET,HSCALE(OBJET)+50,VSCALE(OBJET)+50)
```

L'appel de la fonction sera ensuite fait comme pour toute autre fonction:

```
EXPAND(FIRSTSEL) par exemple.
```

La redéfinition d'une fonction ayant déjà été définie est possible, la nouvelle définition remplace alors l'ancienne. Ceci peut être utile lorsque vous essayez de définir une fonction complexe, il peut alors être nécessaire de faire plusieurs essais avant d'obtenir le bon résultat.

Voir aussi :

```
READDEF
```

## 1.151 Fonction ARexx ABS

ABS(nombre)

Cette fonction renvoie la valeur absolue du nombre donné en argument. Le nombre peut appartenir à une variable.

---

Exemples :

```
ABS(-8)      renvoie 8
ABS(4)       renvoie 4
ABS(N)       renvoie la valeur absolue du nombre contenu dans
              la variable N
```

Rappel : les nombres doivent être du type entier, les variables doivent naturellement être du type numérique.

## 1.152 Fonction ARexx ARC

ARC (x, y, rayon\_horizontal, rayon\_vertical, angle\_début, angle\_fin)

Trace un arc de cercle. Le centre est spécifié par les coordonnées x et y.

La valeur de l'argument angle de début doit être inférieure à celle de l'angle de fin (utilisez éventuellement une valeur négative). Elles doivent être spécifiées en degrés, avec une précision maximale de 1 degré.

Si le placement est réussi la valeur renvoyée correspond au numéro du nouvel objet, sinon la valeur renvoyée est nulle.

Si les coordonnées x et y sont égales à -1, les dimensions des rayons et des angles s'appliquent au mode curseur, lors du placement d'un arc.

Pour connaître les caractéristiques d'un arc existant, utilisez la fonction COORDS.

Exemples :

```
ARC(100,100,25,25,0,90):ARC(100,100,25,25,90,180)   trace un demi-cercle
effet identique à celui de la seule instruction suivante:
```

```
ARC(100,100,25,25,0,180)
```

ou

```
ARC(100,100,25,25,-180,0)           (trace le demi-cercle complémentaire)
```

Pour dessiner une boîte arrondie (x0, y0, x1, y1):

```
DEF ROUNDED_BOX(x0,y0,x1,y1)=
```

```
ARC(x0+10,y0+10,10,10,-90,0):DRAW(x0+10,y0,x1-10,y0):ARC(x1-10,y0 ↔
+10,10,10,0,90):
```

```
DRAW(x1,y0+10,x1,y1-10):ARC(x1-10,y1-10,10,10,90,180):DRAW(x1-10,y1,x0+10,y1):
```

```
ARC(x0+10,y1-10,10,10,180,270):DRAW(x0,y1-10,x0,y0+10)
```

Voir aussi :

menu Dessin/Placer~arc~de~cercle

fonction ELLIPSE

## 1.153 Fonction ARexx ARRAYDIM

ARRAYDIM(tableau)

Cette fonction retourne le nombre d'éléments appartenant à un tableau, s'il existe. Sinon la valeur 0 est renvoyée. Elle peut ainsi servir à tester l'existence d'un tableau de données au début d'un script.

Voir aussi : DIMARRAY, DELARRAY, GETARRAY, SETARRAY

## 1.154 Fonction ARexx ASC

ASC("texte",rang)

Cette fonction renvoie le code ASCII du caractère appartenant à la chaîne de texte passée en argument, du rang spécifié. Le rang doit être compris entre 1 et la longueur de la chaîne passée en argument.

Exemples :

ASC("essai",1) renvoie 101 (code ASCII du caractère e)  
ASC(TEXTE,LEN(TEXTE)) renvoie le code ASCII du dernier caractère de la variable TEXTE.

Voir aussi : CHR

## 1.155 Fonction obsolète

ASK("titre")

Cette fonction a été supprimée dans la version 2.0  
Utilisez en place l'une des fonctions ASKNUM ou ASKTEXT.

Elle peut être émulée à l'aide de la définition suivante:  
DEF ASK(T)=ASKTEXT(T,"")

## 1.156 Fonction ARexx ASKNUM

ASKNUM("titre",nombre)                      Version 1.6

Ouvre une boîte de requête permettant la saisie d'un nombre. Le second argument est utilisé pour initialiser cette demande. Si l'utilisateur clique sur Annule, une chaîne nulle est renvoyée, sinon une chaîne correspondant au nombre saisi est renvoyée. Seuls des nombres entiers peuvent être saisis.

Voir aussi : ASKTEXT.

---

## 1.157 Fonction ARexx ASKTEXT

ASKTEXT("titre", "texte")                      Version 1.6

Affiche une requête permettant la saisie d'une chaîne de caractères.  
Le second paramètre est utilisé pour initialiser la demande.  
Une chaîne nulle est renvoyée si l'utilisateur clique sur Annule.

Exemple:

```
ASKTEXT("Entrez le premier"+CHR(10)+"mot puis pressez"+CHR(10)+"la touche ENTRÉE ↵", "")
```

Voir aussi : ASKNUM.

## 1.158 Fonction ARexx BLINK

BLINK(numéro\_objet)

Provoque le clignotement de l'objet spécifié (trois fois).  
L'argument doit être compris entre 1 et OBJECTS.

## 1.159 Fonction ARexx BOX

BOX(x0, y0, x1, y1)

Trace un rectangle en utilisant les coordonnées spécifiées.  
Le type de ligne tracé dépend du mode alors valide (voir DRAWMODE).

Si le placement du tracé réussit, cette fonction renvoie le numéro de l'objet qui a été placé, sinon elle renvoie 0.

Voir aussi : DRAW.

## 1.160 Fonction ARexx CALL

CALL("script", argument1, argument2...)

Appel d'un script ARexx. Le nom du script, donné en argument, n'a pas à inclure l'extension ".amiCAD". Cette fonction n'est pas utile dans un script ARexx mais peut servir dans une macro-commande (ALT-Fx ou macro locale).

L'appel peut être fait en spécifiant de 0 à 15 arguments (numériques ou chaînes de caractères).

L'exécution du script se fait en asynchrone, c'est à dire que la main est rendue au programme appelant avant la fin de l'exécution du script. La valeur renvoyée est le nom du script (sans grand intérêt)...

---



Exemples :

```
'CALL("EditScript","Zoom")'  
'CALL("multiplier",1.5,X)'
```

## 1.161 Fonction ARexx CHR

CHR(code)

Renvoie le caractère possédant le code ASCII spécifié.

Utilisé pour obtenir un saut de ligne (CHR(10)) ou un caractère spécial non présent sur le clavier. Le code peut varier de 1 à 255 maxi.

Cette fonction peut aussi être utilisée pour écrire des caractères spéciaux, qui ne font pas partie du jeu de caractères de l'Amiga.

Ainsi les caractères suivants sont utilisables:

- 128: symbole de l'amplification (triangle pointé à droite)
- 129: symbole collecteur ouvert
- 130: symbole du OU (supérieur ou égal)
- 131: flèche dirigée à droite
- 133: symbole utilisé pour les entrées actives sur un front (supérieur)
- 134: symbole de l'hystérésis
- 135: symbole "trois états" (haute impédance)
- 136: symbole d'une impulsion
- 137: flèche dirigée à gauche
- 138: symbole décrivant un signal analogique
- 139: lettre correspondant au signe "ohm" (pour la valeur des résistances)
- 140: lettre "alpha" (pour les potentiomètres)
- 141: lettre TAU (constantes de temps)

Essayez ces codes à l'aide d'une macro du type

```
WRITE(CHR(140),100,100)
```

Vous pouvez ajouter un de ces caractères (ou n'importe quel autre à une chaîne de caractères avec une écriture de la forme suivante:

```
"Flèche à droite: "+CHR(131)
```

Voir aussi : ASC

## 1.162 Fonction ARexx CLIPPATH

CLIPPATH("chemin")

Détermine l'emplacement (le tiroir) où sont situés les clips. Cette fonction permet de modifier le chemin acquis à l'aide de l'outil CLIPS.

La fonction renvoie le chemin utilisé avant son utilisation. Si le chemin spécifié est une chaîne nulle, il n'y a pas de modification du chemin utilisé: CLIPPATH(""), seul le

chemin courant est renvoyé.

Exemple :

```
CLIPPATH("Work:AmiCAD/Clips")
```

## 1.163 Fonction ARexx CLIPUNIT

CLIPUNIT(unité)

Cette fonction permet de choisir la zone mémoire utilisée pour les opérations de couper/coller. Ce numéro peut varier de 1 à 10. L'unité 5 est souvent utilisée dans les scripts ARexx, si nécessaire.

La valeur renvoyée correspond au numéro de l'unité qui été utilisée AVANT l'appel à cette fonction. Si vous voulez connaître celui-ci sans en changer passez une valeur négative ou nulle en argument.

Attention : si la valeur renvoyée par cette fonction est nulle, c'est que la fonction a échoué. Cela se produit notamment si une tâche essaie de changer de clip actif alors que ce même clip est en cours d'utilisation par une autre tâche (Version 2.05).

Exemple de script :

```
'CLIPUNIT(2)'          choix d'une nouvelle unité (numéro 2)
clip=RESULT            conserver le numéro de l'ancienne unité
...
'CLIPUNIT('clip')'    revalider l'ancienne unité
```

Voir aussi : COPY, PASTE.

## 1.164 Fonction ARexx CLOSE

CLOSE(fenêtre)

Cette fonction provoque la fermeture de la fenêtre courante (argument positif ou nul) ou de toutes les fenêtres, si l'argument vaut -1. Attention, si le contenu de la fenêtre a été modifié mais non sauvé, les modifications seront perdues, utilisez la fonction MODIF pour savoir s'il y a eu des changements. Aucune valeur n'est renvoyée par cette fonction.

Pour cacher une fenêtre, sans perdre ses informations, utilisez la commande MENU("Cacher"); pour la réduire à sa taille minimale, utilisez la commande MENU("Réduire").

Exemple :

```
CLOSE(0)              Pour fermer la fenêtre courante.
CLOSE(-1)             Ferme toutes les fenêtres.
```

Voir aussi : OPEN, LOAD.

## 1.165 Fonction ARexx COL

COL(numéro\_objet)

Cette fonction permet de connaître le numéro de colonne où est situé l'objet spécifié.

L'argument doit être compris entre 1 et OBJECTS, s'il est nul c'est la position du bord gauche du bloc sélectionné qui est renvoyée (ou -1 s'il n'y a aucun élément sélectionné).

Voir aussi: ENDCOL, LINE, COORDS.

## 1.166 Fonction ARexx CONVERT

CONVERT(numéro\_objet, 0/1/-1)

Le premier argument doit être compris entre 1 et OBJECTS.

S'il est nul, l'action validera le mode de placement courant des composants (comme en utilisant le menu Dessin/Placer composant par exemple). La valeur renvoyée sera alors égale à 0 si le mode de placement courant était le mode "normal", 1 si le mode "alterné" était valide, la valeur renvoyée dépend du mode en vigueur avant l'appel de la fonction (version 1.5).

Cette fonction permet de choisir le type de symbole pour un composant ou une jonction. Selon la valeur du second argument, l'action sera la suivante :

- égal à 0: c'est le symbole normal qui sera utilisé,
- égal à 1: c'est le deuxième symbole qui sera utilisé,
- égal à -1: on change de symbole.

La valeur renvoyée est alors égale à 0 ou 1 si l'objet sélectionné est bien un composant, sinon c'est un -1.

Voir aussi: menu Dessin/Alterné~symbole, READCONV

## 1.167 Fonction ARexx COORDS

COORDS(numéro\_objet)

Cette fonction retourne les coordonnées de l'objet spécifié sous forme de chaîne de caractères, séparées par des virgules.

Si l'objet est une ligne ou une boîte les coordonnées sont de la forme x0,y0,x1,y1,

si c'est un triangle elles sont de la forme

x0,y0,x1,y1,x2,y2,

si c'est un arc elles sont de la forme

x,y,rayon\_horizontal,rayon\_vertical,angle\_début,angle\_fin

si c'est une ellipse elles sont de la forme  
x,y,rayon\_horizontal,rayon\_vertical  
alors qu'elles sont de la forme x,y pour tout autre objet.

L'argument doit être compris entre 1 et OBJECTS.

Exemple d'utilisation :

```
'COORDS('ligne')'; coord=result  
/* retrouver les différentes coordonnées */  
PARSE VAR coord x0 ',' y0 ',' x1 ',' y1
```

Voir aussi : COL, LINE, ENDCOL, ENDLIN.

## 1.168 Fonction ARexx COPY

COPY(clip)

Cette fonction permet de recopier les objets sélectionnés dans la zone mémoire spécifiée. Le numéro de clip peut varier de 1 à 10.

Utilisez la fonction PASTE pour coller les objets ainsi collés ou choisissez l'unité comme unité par défaut (CLIPUNIT) puis utilisez le menu Édition/Coller.

La valeur renvoyée est égale à 1 si tout se passe normalement.

## 1.169 Fonction ARexx CURSMODE

CURSMODE(type)

Version 2.00

Cette fonction permet de placer le curseur dans un mode spécifié par son argument:

- si type est égal à 0: mode placement clip,
- si type est égal à 1: mode placement composant,
- si type est égal à 2: mode placement fils,
- si type est égal à 3: mode placement arcs de cercles,
- si type est égal à 4: mode placement texte,
- si type est égal à 7: mode placement jonctions,
- si type est égal à 10: mode placement ellipse,
- si type est égal à 22: mode placement rectangle.
- si type est égal à 26: mode placement triangle.

Toute autre valeur annule le mode courant.

La valeur renvoyée est celle correspondant au mode en vigueur avant l'exécution de la fonction.

Pour passer en mode de placement d'un texte ou d'un connecteur utilisez les fonctions WRITE, INPUT ou OUTPUT.

## 1.170 Fonction ARexx DATE

DATE(jour)

Renvoie la date courante. Si l'argument jour est différent de zéro, le jour de la semaine est inclus.

Exemples :

DATE(0)           renvoie 18-Oct-97

DATE(1)           renvoie Samedi 18-Oct-97

Voir aussi : TIME.

## 1.171 Fonction ARexx DELARRAY

DELARRAY(tableau)

Cette fonction permet de libérer la mémoire utilisée pour gérer un tableau de données. Si le tableau n'existe pas aucune erreur n'est générée, la fonction retourne alors 0 (sinon la valeur renvoyée est le nombre d'éléments qui avaient été déclarés).

Voir aussi: ARRAYDIM, DIMARRAY, GETARRAY, SETARRAY

## 1.172 Fonction ARexx DELETE

DELETE(numéro\_objet)

Supprime l'objet dont le numéro est spécifié.  
Renvoie le nombre d'objets restants.

La valeur de l'argument doit être comprise entre 1 et OBJECTS, si l'argument est nul, tous les éléments alors sélectionnés dans la fenêtre courante sont effacés, si l'argument est négatif (normalement égal à -1), TOUS les éléments du document courant sont supprimés (version 2.01).

Attention: cette fonction peut modifier les numéros des objets restants.

## 1.173 Fonction ARexx DEVPINS

DEVPINS(numéro\_objet)

Cette fonction est utilisée afin d'élaborer une netlist.

La valeur de l'argument doit être comprise entre 1 et OBJECTS.

L'objet spécifié doit être un composant, la fonction renvoie alors le nombre de connexions que possède ce composant, sinon un message d'erreur est affiché.

---

Voir aussi : PINCOL, PINLINE.

### 1.174 Fonction ARexx DIMARRAY

DIMARRAY(tableau, nombre\_éléments)

Cette fonction permet de réserver la place en mémoire pour gérer un tableau de données. Les données pouvant ensuite y être stockées peuvent être du type nombre (entiers) ou chaînes de caractères, la première affectation déterminant le type de données à utiliser, comme pour les variables. Le nombre d'éléments doit être supérieur à 1 et inférieur ou égal à  $2^{24}$  (16777216).

Voir aussi: ARRAYDIM, GETARRAY, DELARRAY, SETARRAY

### 1.175 Fonction ARexx DIMSHEET

DIMSHEET(largeur, hauteur)

Cette fonction permet de changer les dimensions de la fenêtre de travail courante. Ces dimensions sont données en pixels. Utilisez le menu Projet/Informations pour connaître les dimensions de la feuille de travail.

Les dimensions maximales pouvant être prises par la fenêtre dépendent bien sûr de la quantité de mémoire chip installée sur votre système. Des dimensions de 1100 par 700 sont convenables pour un système équipé de 2 Mo de chip.

Valeur renvoyée: 1 si l'opération a réussi, 0 dans le cas contraire. Attention, aucun contrôle n'est fait sur les valeurs passées en arguments.

Voir aussi: menu Préférences/Dimensions document, fonctions WWIDTH, WHEIGHT

### 1.176 Fonction ARexx DRAW

DRAW(x0, y0, x1, y1)

Trace une ligne en utilisant les coordonnées spécifiées. Le type de ligne tracé dépend du mode alors valide (voir DRAWMODE).

Si le placement du tracé réussit, cette fonction renvoie le numéro de l'objet qui a été placé, sinon elle renvoie 0.

Voir aussi : BOX, COORDS.

---

## 1.177 Fonction ARexx DRAWMODE

`DRAWMODE(type_ligne)`

Détermine quel sera le type de ligne qui sera tracé (voir DRAW).  
Si `type_ligne` est égal à 0: ce seront des lignes en pointillés,  
égal à 1: ce seront des lignes "normales" (liaisons),  
égal à 2: ce seront des lignes doubles,  
égal à 3: ce seront des bus.  
Si `type_ligne` est inférieur à 0 et supérieur à -256, ce seront des lignes de largeur personnalisée qui seront tracées, avec une largeur égale à la valeur absolue de celle qui a été spécifiée (v1.1).

Le menu Dessin est mis à jour en fonction du type de tracé sélectionné.

Cette fonction renvoie le type de ligne qui était en vigueur AVANT l'appel de la fonction.

## 1.178 Fonction ARexx EDIT

`EDIT(numéro_objet)`

Cette fonction appelle la requête permettant d'éditer un élément du schéma. Elle a le même effet qu'un double clic sur un objet à l'aide du bouton gauche de la souris, ou d'appuyer sur la touche ENTRÉE après avoir sélectionné un objet.  
Aucune valeur n'est renvoyée.

La valeur de l'argument doit être comprise entre 1 et OBJECTS.

Cette fonction est utile associée à une combinaison de touches du clavier.

Exemple :

```
MAP("alt-e", "EDIT(FIRSTSEL)")
```

Il suffit alors de sélectionner un objet puis de faire la combinaison de touches ALT-e au clavier pour appeler la requête d'édition du premier élément sélectionné.

## 1.179 Fonction ARexx ELLIPSE

`ELLIPSE(x, y, rayon_horizontal, rayon_vertical)`

Trace une ellipse, le centre est spécifié par les coordonnées `x` et `y`, les rayons par les deux arguments suivants. Spécifiez deux rayons égaux pour tracer un cercle ou utilisez la macro suivante :

```
DEF CERCLE(x,y,rayon)=ELLIPSE(x,y,rayon,rayon)
```

Si le placement réussit, le numéro du nouvel objet est renvoyé, sinon c'est un 0.

---

Si les coordonnées x et y sont égales à -1, les dimensions des rayons s'appliquent au mode curseur, lors du placement d'une ellipse.

Utilisez la fonction COORDS pour lire les caractéristiques d'une ellipse existante.

Voir aussi : ARC

## 1.180 Fonction ARexx ENDCOL

ENDCOL(numéro\_objet)

Cette fonction renvoie l'abscisse du point déterminant la fin du tracé d'une ligne (pointillés ou largeur quelconque). Si l'objet est d'un type différent (composant, jonction ou autre) la valeur renvoyée n'a aucune signification.

Le numéro d'objet doit avoir sa valeur comprise entre 1 et OBJECTS.

Voir aussi : COL, LINE, ENDLINE, COORDS.

## 1.181 Fonction ARexx ENDLINE

ENDLINE(numéro\_objet)

Cette fonction renvoie l'ordonnée du point déterminant la fin du tracé d'une ligne (pointillés ou largeur quelconque). Si l'objet est d'un type différent (composant, jonction ou autre) la valeur renvoyée n'a aucune signification.

Le numéro d'objet doit avoir sa valeur comprise entre 1 et OBJECTS.

Voir aussi : COL, LINE, ENDCOL, COORDS.

## 1.182 Fonction ARexx EXEC

EXEC("chaîne de caractères")

Demande l'interprétation et l'exécution de la chaîne de caractères passée en argument, comme s'il s'agissait d'une ligne de commande.

Le résultat dépend naturellement du contenu de la chaîne passée en argument.

Exemple :

---



EXEC(READTEXT(OBJET))    interprète la ligne de texte associée à l'objet ←  
                              spécifié  
EXEC(READMAP("CTRL-"))    exécute la séquence associée à la combinaison de ←  
                              touches CTRL-)  
EXEC(READMAP("F5"))        exécute la séquence associée à la touche F5.

L'ancienne fonction peut être remplacée en utilisant une  
définition de fonction (dans le fichier startup, par exemple):

```
DEF EXECKEY(X)=EXEC(READMAP(X))  
il devient alors possible d'écrire :  
    EXECKEY("F5")  
    ou  
    EXECKEY("Alt-F5")
```

### 1.183 Fonction ARexx FILENAME

FILENAME(nom)

Renomme la fenêtre courante, avec le nom spécifié.  
Si une chaîne nulle ("") est passée en argument, cette  
fonction renvoie le nom de la fenêtre, y compris le  
chemin complet.

Voir aussi : FILEPART

### 1.184 Fonction ARexx FILEPART

FILEPART("nom")

Renomme la fenêtre courante, en conservant le chemin.  
Si une chaîne nulle ("") est passée en argument, cette  
fonction renvoie le nom de la fenêtre courante, sans le  
chemin complet.

Exemple :

Supposons que le fichier courant soit "RAM:sources/Schéma"  
la fonction FILEPART("") renverrait Schéma, alors que la  
fonction FILEPART("Nouveau schéma") renommerait le fichier  
"RAM:sources/Nouveau schéma".

Voir aussi : FILENAME

### 1.185 Fonction ARexx FINDLAB

FINDLAB(premier\_objet,"Label")

Cette fonction permet de trouver un fil possédant le label  
spécifié. La recherche commence à partir du numéro d'objet  
spécifié. La valeur du premier argument doit être compris  
entre 1 et OBJECTS.

---

La valeur renvoyée est égale au numéro de l'objet qui a été trouvé, ou à 0 si la recherche n'a pas abouti.

La recherche s'effectue sans tenir compte de la casse des lettres, c'est à dire que les majuscules et les minuscules ne sont pas différenciées. D'autre part, vous pouvez inclure des "jokers" (ou caractères génériques) dans la chaîne recherchée (voir documentation AmigaDOS pour les diverses possibilités).

Nota: c'est le numéro du fil qui est renvoyé, pas celui de son label (utilisez GETLABEL pour avoir celui-ci).

Exemples :

```
FINDLAB(1,"0")  Renvoie le numéro du premier fil possédant le label 0
FINDLAB(1,"Lab#?") Renvoie le numéro du premier label commençant par
                  les lettres 'Lab'
```

## 1.186 Fonction ARexx FINDLINE

FINDLINE(premier\_objet,x,y)                      Version 1.4

Lance une recherche pour savoir si une ligne passe aux coordonnées spécifiées, à partir du numéro d'objet spécifié. La valeur du premier argument doit être compris entre 1 et OBJECTS.

La valeur renvoyée est égale au numéro de l'objet qui a été trouvé, ou à 0 si la recherche n'a pas abouti. Si les coordonnées spécifiées sont celles de l'extrémité de la ligne qui a été trouvée, son numéro est renvoyé sous forme négative.

Attention: seules les lignes verticales ou horizontales sont prises en compte, les lignes obliques ne sont pas testées dans cette version (1.4), sauf pour leurs extrémités. D'autre part seules les lignes de largeur égale à 1 sont testées. Pour différencier une ligne servant de cadre et une ligne servant de liaison, tracez les cadres en utilisant le mode "Largeur quelconque" et les liaisons en ne validant aucun mode de tracé.

Exemple :

```
pour savoir si une ligne passe à un point de coordonnées x0,y0:
'ABS(FINDLINE(1,'x0','y0'))'; l=result
if l>0 then présence_ligne
```

Voir aussi : FINDOBJ

## 1.187 Fonction ARexx FINDOBJ

FINDOBJ(premier\_objet,type,x,y)                      Version 1.4

Lance la recherche du premier objet situé aux coordonnées

---

spécifiées.

Le premier argument sert à commencer la recherche après un objet déjà trouvé. Cet argument doit prendre une valeur comprise entre 1 (la recherche s'effectuera sur tous les objets du schéma) et OBJECTS.

Le second argument précise quel est le type d'objet recherché, il doit correspondre à la valeur renvoyée par la fonction TYPE. S'il est nul, le premier objet trouvé sera renvoyé, quel que soit son type.

Les coordonnées doivent correspondre au point d'ancrage pour les composants (c'est à dire le point où était situé le curseur au moment où il a été posé), à une extrémité pour les lignes ou au centre pour les ellipses ou jonctions. Si l'une des coordonnées est négative, elle n'est pas prise en compte. Si les deux le sont, aucune ne sera prise en compte, c'est à dire que l'objet sera testé sans tenir compte de son emplacement.

La valeur renvoyée est égale au numéro de l'objet qui a été trouvé, ou à 0 si la recherche n'a pas abouti.

Exemples :

FINDOBJ(1,1,-1,-1) renvoie le numéro du premier composant.

FINDOBJ(1,2,100,200) renvoie le numéro d'une ligne ayant l'une de ses extrémités en 100,200, si il y en a une.

FINDOBJ(1,7,50,-1) renvoie le numéro de la première jonction située sur la colonne 50.

Voir aussi : FINDLINE

## 1.188 Fonction ARexx FINDPART

FINDPART(premier\_objet, "composant")

Lance la recherche du composant dont le nom est spécifié sur le document courant.

Ce nom correspond au nom du symbole, "RÉSISTANCE" par exemple pour une résistance.

Le premier argument sert à commencer la recherche après un objet déjà trouvé. Cet argument doit prendre une valeur comprise entre 1 (la recherche s'effectuera sur tous les objets du schéma) et OBJECTS.

La recherche s'effectue sans tenir compte de la casse des lettres, c'est à dire que les majuscules et les minuscules ne sont pas différenciées. D'autre part, vous pouvez inclure des "jokers" (ou caractères génériques) dans la chaîne recherchée (voir documentation AmigaDOS pour les diverses possibilités).

La valeur renvoyée est égale au numéro de l'objet qui a été trouvé, ou à 0 si aucun objet ne correspond.

---

Exemples :

```
'FINDPART(1,"RÉSISTANCE")'; objet=result  
'FINDPART(1,"(Rés|Cond)#?")'
```

## 1.189 Fonction ARexx FINDREF

FINDREF(premier\_objet, "référence")

Lance la recherche du composant dont la référence est spécifiée, R3 par exemple pour une résistance ou CI9 pour un circuit intégré.

Le premier argument sert à commencer la recherche après un objet déjà trouvé. Cet argument doit prendre une valeur comprise entre 1 (la recherche s'effectuera sur tous les objets du schéma) et OBJECTS.

La recherche s'effectue sans tenir compte de la casse des lettres, c'est à dire que les majuscules et les minuscules ne sont pas différenciées. D'autre part, vous pouvez inclure des "jokers" (ou caractères génériques) dans la chaîne recherchée (voir documentation AmigaDOS pour les diverses possibilités).

La valeur renvoyée est égale au numéro de l'objet qui a été trouvé, ou à 0 si aucun objet ne correspond.

Nota: c'est le numéro du composant qui est renvoyé, pas celui de sa référence (utilisez GETREF pour avoir celui-ci).

Exemples :

```
'FINDREF(1,"R4")'; objet=result  
'FINDREF(1,"R#?")'; objet=result  
IF(O=FINDREF(1,"C2"),BLINK(O),0)
```

Voir aussi : FINDVAL

## 1.190 Fonction ARexx FINDVAL

FINDVAL(premier\_objet, "valeur")

Lance la recherche du composant dont la valeur ou le type est spécifié, 10k par exemple pour une résistance ou 7400 pour un circuit intégré.

Le premier argument sert à commencer la recherche après un objet déjà trouvé. Cet argument doit prendre une valeur comprise entre 1 (la recherche s'effectuera sur tous les objets du schéma) et OBJECTS.

La recherche s'effectue sans tenir compte de la casse des lettres, c'est à dire que les majuscules et les minuscules ne sont pas différenciées. D'autre part, vous pouvez inclure des "jokers" (ou caractères génériques) dans la chaîne recherchée (voir documentation AmigaDOS pour les diverses possibilités).

---

La valeur renvoyée est égale au numéro de l'objet qui a été trouvé, ou à 0 si aucun objet ne correspond.  
Nota: c'est le numéro du composant qui est renvoyé, pas celui de sa valeur (utilisez GETVAL pour avoir celle-ci).

Exemples :

```
'FINDVAL(1,"10k")'; objet=result  
IF (O=FINDVAL(1,"10$\mathrm{\mu}$F"),MARK(O),0)
```

Voir aussi : FINDREF

## 1.191 Fonction ARexx FIRSTSEL

FIRSTSEL

Cette fonction renvoie le numéro du premier objet sélectionné. Elle ne nécessite aucun argument. Si aucun objet n'est marqué la valeur renvoyée est nulle.

Voir aussi : NEXTSEL

## 1.192 Fonction ARexx FONTNAME

FONTNAME      Version 2.05 (argument supprimé)

Renvoie le nom de la fonte utilisée dans l'écran courant.

Voir aussi : FONTSIZE.

## 1.193 Fonction ARexx FONTSIZE

FONTSIZE      Version 2.05 (argument supprimé)

Renvoie la taille de la fonte utilisée dans l'écran courant.

Voir aussi : FONTNAME.

## 1.194 Fonction ARexx FOR

FOR(init,condition\_fin,action1,...)

Cette fonction permet de définir des boucles. Le premier argument (init) est exécuté une seule fois, quand l'appel de la fonction vient d'être fait. Le second argument définit la condition de fin de boucle. Enfin le troisième argument ainsi que les arguments suivants, s'ils existent, sont évalués à chaque exécution de la

boucle.

L'utilisation de cette fonction permet souvent un gain de temps important dans l'exécution d'un script, le nombre de messages pouvant diminuer de façon significative si elle est utilisée à bon escient.

Exemples :

```
FOR (I=0, I<10, I=I+1)
```

Dans cet exemple la variable I est initialisée à la valeur 0, tant que cette valeur est inférieure à 10, on incrémente cette valeur. I va donc prendre successivement les valeurs 1 à 10.

```
FOR (I=10, I>0, DELETE(I), I=I-1)
```

Cette formule comprend une instruction supplémentaire permettant de supprimer les 10 premiers objets.

```
N=100:FOR (I=0:J=0, I<=N, J=J+I, I=I+1):J
```

Cette formule permet de calculer la somme des 100 premiers nombres. Le résultat de la somme est renvoyé (:J à la fin).

Remarque : une boucle bloquée peut être interrompue par deux moyens: soit par le nombre de boucles maximal (défini par la fonction SECURITY, soit en appuyant simultanément sur les trois touches CTRL, ALT et ESC.

Voir aussi : WHILE.

## 1.195 Fonction ARexx GETARRAY

GETARRAY(tableau, indice)

Cette fonction retourne la valeur d'une variable appartenant à un tableau. L'indice doit être compris et 0 et la valeur maximale déterminant la taille du tableau (diminuée de 1).

Voir aussi: ARRAYDIM, DIMARRAY, DELARRAY, SETARRAY

## 1.196 Fonction ARexx GETCOLOR

GETCOLOR(couleur)

Cette fonction permet de connaître les niveaux RVB d'une couleur donnée. Le numéro de la couleur peut varier de 0 à 15. Les niveaux sont renvoyés sous la forme \$rrr,\$vvv,\$bbb dans une chaîne de caractères. Cette chaîne est de la même forme que celle qui est rencontrée dans le fichier des préférences.

Voir aussi : SETCOLOR.

---

## 1.197 Fonction ARexx GETDEVS

GETDEVS("nom\_circuit")            Version 1.4

Cette fonction permet de savoir combien de portes contient un circuit de bibliothèque donné. Le nom du circuit doit correspondre à celui d'un symbole présent en bibliothèque, il peut être spécifié sans tenir compte des minuscules ou des majuscules.

Si le composant existe mais sans numérotation de bornes (comme une résistance) la valeur 0 est renvoyée.

Si le composant n'existe pas dans les bibliothèques alors présentes en mémoire, le message "Nom argument incorrect" est affiché.

Exemples :

```
GETDEVS("LM324")        renvoie 4 (4 amplis)
GETDEVS("4011")        renvoie 4 (4 portes NAND)
GETDEVS("4017")        renvoie 1 (1 compteur)
```

Voir aussi : READDEV, SETDEV.

## 1.198 Fonction ARexx GETLABEL

GETLABEL(numéro\_objet)

Renvoie le numéro du label associé au fil dont le numéro est passé en argument.

Le numéro de l'objet passé en argument doit naturellement être un fil.

Si l'objet est d'un autre type la fonction renvoie -1.

Si ce label n'existe pas la fonction renvoie 0.

Le numéro d'objet doit avoir sa valeur comprise entre 1 et OBJECTS.

Exemple : READTEXT(GETLABEL(FIRSTSEL))

Voir aussi : SETLABEL, FINDLAB, LINKLAB, UNLINK

## 1.199 Fonction ARexx GETLINE

GETLINE("titre1", "titre2")

Cette fonction permet de définir une ligne à l'aide de la souris. Le premier titre est affiché dans la barre de titre pendant l'attente du premier clic sur le bouton gauche de la souris par l'utilisateur. Quand ceci a été fait, le second titre est affiché à son tour, et une ligne est dessinée au fur et à mesure des déplacements de la souris, jusqu'à un second clic.

Toute action sur une touche du clavier ou un autre bouton de la souris entraîne l'annulation de la fonction (elle renvoie alors une chaîne nulle).

---

Si la fonction a réussi, elle renvoie une chaîne de caractères définissant les coordonnées de la ligne définie, comme pour la fonction COORDS.

Le tracé est effectué en tenant compte des modes définis pour le tracé: orthogonal, alignement sur la grille ou non.

Voir aussi : SETORTHO, SNAPGRID, GETZONE.

## 1.200 Fonction ARexx GETNET

GETNET(numéro\_objet)

Cette fonction renvoie le numéro de l'objet ligne associé à un label. Le numéro d'objet doit naturellement être celui d'un label, sinon la valeur -1 est renvoyée.

Voir aussi : GETLABEL, SETLABEL, LINKLAB, FINDLAB

## 1.201 Fonction obsolète

GETPART("composant")

Cette fonction a été supprimée dans la version 2.0  
Utilisez en place la fonction PUTPART.

Vous pouvez aussi utiliser la définition suivante (dans le fichier startup.AmiCAD par exemple, voir type d'outil STARTUP):  
DEF GETPART(C)=PUTPART(C,-1,-1)

## 1.202 Fonction ARexx GETPOINT

GETPOINT("titre")                      Version 1.6

Cette fonction affiche le message spécifié dans la barre de titre et renvoie les coordonnées du point où clique l'utilisateur. Ces coordonnées sont renvoyées sous la forme colonne\*32768+ligne. Pour retrouver l'abscisse et l'ordonnée, vous pouvez utiliser l'écriture suivante dans un script ARexx:  
'GETPOINT("Cliquez sur un point")'  
p=result  
col=bitand(p,'FFFF0000'x)  
col=col/32768  
ligne=bitand(p,'FFFF'x)

Si l'utilisateur clique sur le bouton droit ou sur une touche du clavier cette fonction renvoie la valeur -1.

Les macros suivantes peuvent être définies afin de déterminer les numéros de ligne et de colonne, à partir du résultat de cette

---



```
fonction, directement sous AmiCAD:  
DEF LIGNE(P)=P&0xFFFF  
DEF COLONNE(P)=P>>15
```

## 1.203 Fonction ARexx GETPOS

GETPOS (numéro\_objet)

Renvoie une valeur déterminant le mode de placement d'un composant, d'un texte ou d'un connecteur. Si l'objet occupe une position normale (aucune rotation), cette valeur est égale à 1, sinon elle peut prendre les valeurs 2 à 4 en fonction du nombre de rotations subi par l'objet spécifié. Si l'objet spécifié a subi une symétrie, la valeur renvoyée est négative.

Si l'objet spécifié est d'un autre type, la valeur renvoyée est toujours nulle.

Voir aussi : ROTATE, SYMMETRY.

## 1.204 Fonction ARexx GETREF

GETREF (numéro\_objet)

Renvoie le numéro de l'objet associé à la référence du composant spécifié.

Le numéro de l'objet passé en argument doit naturellement être un composant.

Si l'objet est d'un autre type la fonction renvoie -1.

Si cette référence n'existe pas la fonction renvoie 0.

Le numéro d'objet doit avoir sa valeur comprise entre 1 et OBJECTS.

Exemple : READTEXT (GETREF (FIRSTSEL))

Voir aussi : SETREF, LINKREF, UNLINK, GETVAL

## 1.205 Fonction ARexx GETVAL

GETVAL (numéro\_objet)

Renvoie le numéro de l'objet associé à la valeur du composant spécifié.

Le numéro de l'objet passé en argument doit naturellement être un composant.

Si l'objet est d'un autre type la fonction renvoie -1.

Si cette référence n'existe pas la fonction renvoie 0.

---

Le numéro d'objet doit avoir sa valeur comprise entre 1 et OBJECTS.

Exemple : READTEXT(GETVAL(FIRSTSEL))

Voir aussi : SETVAL, LINKVAL, UNLINK, GETREF

## 1.206 Fonction ARexx GETZONE

GETZONE("titre")                      Version 2.05 (Second argument supprimé)

Cette fonction permet de définir une zone rectangulaire à l'aide de la souris. Le titre est affiché dans la barre de titre pendant la durée de l'opération. Pour définir la zone, l'utilisateur doit cliquer sur le bouton gauche de la souris, maintenir celui-ci appuyé puis le relâcher quand la zone a été définie.

Toute action sur une touche du clavier ou un autre bouton de la souris entraîne l'annulation de la fonction (elle renvoie alors une chaîne nulle).

Si la fonction a réussi, elle renvoie une chaîne de caractères définissant les coordonnées de la zone définie, comme pour la fonction COORDS.

Voir aussi : GETLINE.

## 1.207 Fonction ARexx GROUP

GROUP(numéro\_objet1,numéro\_objet2...)                      Version 1.4

Crée un nouveau groupe comprenant tous les objets spécifiés. Renvoie le numéro du groupe créé.

GROUP(numéro\_objet)                      Variante avec un seul argument  
Renvoie le numéro du groupe dans lequel est inclus l'objet spécifié, ou 0 si pas d'appartenance à un groupe.

Le numéro de chacun des objets doit avoir sa valeur comprise entre 1 et OBJECTS.

Voir aussi : UNGROUP.

## 1.208 Fonction ARexx HEIGHT

HEIGHT (numéro\_objet)

Renvoie la hauteur de l'objet spécifié, en pixels.

Le numéro d'objet doit avoir sa valeur comprise entre 1 et OBJECTS, s'il est nul c'est la hauteur du bloc sélectionné qui est renvoyée (ou -1 s'il n'y a aucun élément sélectionné).

---

Voir aussi : WIDTH

## 1.209 Fonction ARexx HELP

```
HELP("node")
```

Cette fonction permet l'appel d'AmigaGuide, comme par le menu Projet/Aide. L'argument doit être le nom d'un noeud (node) du fichier AmiCAD.guide. Vous pouvez, en particulier, spécifier n'importe quel nom de fonction ou de menu.

Exemples :

```
HELP("Copy~to~clip")
```

```
HELP("DRAW")
```

## 1.210 Fonction ARexx HSCALE

```
HSCALE(numéro_objet)
```

Cette fonction renvoie la valeur de l'échelle horizontale de l'objet spécifié. La valeur de l'argument doit être comprise entre 1 et OBJECTS.

Si l'argument est nul c'est la valeur de placement courante qui est renvoyée (v2.07).

Voir aussi : VSCALE, SETSCALE.

## 1.211 Fonction ARexx IF

```
IF(x, a1, a2)
```

Si x est différent de zéro, renvoie a1 sinon renvoie a2.

Seul a1 OU a2 sera évalué, selon le résultat de x. À noter que les arguments a1 et a2 peuvent être de n'importe quel type, ils peuvent également faire appel à d'autres fonctions, y compris d'autres fonctions IF.

Exemples :

```
IF(A>B,A,B)                      renvoie la valeur maximale de A et B
```

```
DEF MIN(A,B)=IF(A<B,A,B)        définition fonction MIN
```

```
IF(PUTPART(ASKTEXT("Composant?",""),-1,-1),MENU("Placer~composant"),0)
```

## 1.212 Fonction ARexx INIT

```
INIT(variable,...)
```

Cette fonction est identique à la fonction RESET, cependant le type est réinitialisé, c'est à dire que la variable pourra ensuite prendre n'importe quel type autorisé (numérique ou chaîne de caractères).  
Le nombre d'arguments est quelconque.

Exemple :

```
A=1          La variable A est du type numérique
...
INIT(A)       A est réinitialisée (elle perd sa valeur et son type)
...
A="Chaîne de caractères"    La variable A est maintenant du type chaîne
```

Voir aussi : RESET

### 1.213 Fonction ARexx INPUT

```
INPUT("nom", x, y)
```

Place le connecteur avec le nom spécifié, aux coordonnées x, y.  
La flèche du connecteur est dirigée vers la gauche (connecteur d'entrée), sauf si vous utilisez la fonction SYMMETRY, auquel cas elle est dirigée vers la droite.  
Si les valeurs de x et de y sont nulles, la chaîne spécifiée est placée sous le curseur, celui-ci est placé en mode placement de connecteur. L'utilisateur peut alors cliquer où il veut pour le placer (version 1.3).

Si la fonction réussit, elle renvoie le numéro de l'objet, sinon 0.

L'échelle verticale et l'échelle horizontale courantes sont prises en compte pour déterminer les dimensions de cet élément (voir SETSCALE, ROTATE, SYMMETRY).

Voir aussi : OUTPUT

### 1.214 Fonction ARexx JUNCTION

```
JUNCTION(x, y)
```

Place une jonction à l'emplacement spécifié. L'échelle verticale et l'échelle horizontale courantes sont prises en compte pour déterminer le diamètre de cet élément (voir SETSCALE).

Si le placement est réussi, la fonction renvoie le numéro du nouvel objet qui a été placé, sinon elle renvoie 0.

## 1.215 Fonction ARexx LANGUAGE

LANGUAGE      Version 2.07

Renvoie le nom de la "locale" utilisée par le système.  
Pour le français la chaîne "français.language" est renvoyée.  
Cette fonction permet notamment d'internationaliser les scripts.

Exemple d'utilisation:

```
'LANGUAGE'  
if result='français.language' then fr=1  
else fr=0  
...  
if fr=1 then 'MESSAGE("Attention, si vous continuez"+CHR(10)+"vous allez perdre ↵  
vos données !")'  
else 'MESSAGE("Warning, if you continue you are"+CHR(10)+"going to loose all ↵  
your work !")'
```

## 1.216 Fonction ARexx LEN

LEN("chaîne")

Renvoie la longueur de la chaîne de caractères passée  
en argument.

## 1.217 Fonction ARexx LIBRARY

LIBRARY (numéro\_objet)

Renvoie le nom de la bibliothèque à laquelle appartient l'objet  
spécifié, s'il s'agit bien d'un composant, dans le cas contraire  
une chaîne nulle est renvoyée.  
Le nom comprend le chemin complet.

Le numéro d'objet doit avoir sa valeur comprise entre 1  
et OBJECTS.

Voir aussi : PARTNAME, PUTPART.

## 1.218 Fonction ARexx LIBSPATH

LIBSPATH("chemin")

Détermine l'emplacement (le tiroir) où sont situés les bibliothèques  
de symboles.  
Cette fonction permet de modifier le chemin acquis à l'aide de  
l'outil LIBS.

La fonction renvoie le chemin utilisé avant son utilisation.

---

Si le chemin spécifié est une chaîne nulle, il n'y a pas de modification du chemin utilisé: `LIBSPATH("")`. Ceci permet de connaître l'emplacement des bibliothèques.

Exemple (extrait de script ARexx):

```
'LIBSPATH("Work:AmiCAD/Symboles")'; ancienchemin=result  
...  
'LIBSPATH("'ancienchemin'")'      /* remet le chemin initial */
```

## 1.219 Fonction ARexx LINE

`LINE(numéro_objet)`

Cette fonction renvoie le numéro de ligne où est situé l'objet spécifié. L'argument doit être compris entre 1 et `OBJECTS`, s'il est nul c'est la position du bord haut du bloc sélectionné qui est renvoyée (ou -1 s'il n'y a aucun élément sélectionné).

Voir aussi: `ENDLINE`, `COL`, `COORDS`.

## 1.220 Fonction ARexx LINKLAB

`LINKLAB(objet1,objet2)`

Cette fonction permet d'affecter un label à un fil. L'objet déterminant le label doit être du type texte et ne doit pas être déjà lié à un autre composant.

Les arguments doivent être compris entre 1 et `OBJECTS`, ils peuvent être placés dans n'importe quel ordre.

Exemple : `LINKLAB(FIRSTSEL,NEXTSEL(FIRSTSEL))`

Voir aussi: `FINDLAB`, `GETLABEL`, `SETLABEL`, `UNLINK`

## 1.221 Fonction ARexx LINKREF

`LINKREF(objet1,objet2)`

Cette fonction permet d'affecter à un objet du type composant une référence. L'objet déterminant la référence doit être du type texte et ne doit pas être déjà lié à un autre composant. Les arguments doivent être compris entre 1 et `OBJECTS`, ils peuvent être placés dans n'importe quel ordre.

Exemple : `LINKREF(FIRSTSEL,NEXTSEL(FIRSTSEL))`

Voir aussi: `SETREF`, `GETREF`, `LINKVAL`, `SETVAL`, `GETVAL`

---

## 1.222 Fonction ARexx LINKVAL

LINKVAL(objet1,objet2)

Cette fonction permet d'affecter à un objet du type composant une valeur. L'objet déterminant cette valeur doit être du type texte et ne doit pas être déjà lié à un autre composant. Les arguments doivent être compris entre 1 et OBJECTS, ils peuvent être placés dans n'importe quel ordre.

Exemple : LINKVAL(FIRSTSEL,NEXTSEL(FIRSTSEL))

Voir aussi: SETREF, GETREF, LINKREF, SETVAL, GETVAL

## 1.223 Fonction ARexx LOAD

LOAD("nom\_fichier")

Charge le fichier schéma spécifié dans la fenêtre courante. La fenêtre perd son contenu, même s'il avait été modifié. Utilisez la commande MODIF pour savoir si le schéma a été modifié auparavant, sans avoir été sauvé. La fenêtre prend le nom du fichier qui a été chargé.

Renvoie 0 si tout s'est bien passé, sinon un code d'erreur.

Voir aussi : OPEN, SAVE

## 1.224 Fonction ARexx LOADCLIP

LOADCLIP(unité\_clip,"fichier")

Charge le fichier spécifié dans l'unité spécifiée. Si le nom du fichier n'est pas spécifié, la recherche de celui-ci s'effectue dans le chemin courant, puis dans le chemin spécifié par le type d'outil CLIPS.

Attention: le clip n'est pas pour autant placé sous le curseur ou sur le schéma, utilisez pour cela le menu Édition/Coller ou la fonction PASTE.

Voir aussi: CLIPUNIT, SAVECLIP.

## 1.225 Fonction ARexx LOADKEYS

LOADKEYS("fichier\_macros")

Cette fonction a été supprimée dans la version 2.05.

---

## 1.226 Fonction ARexx LOADLIB

```
LOADLIB("nom_bibliothèque")
```

Cette fonction permet de charger une bibliothèque de symboles en mémoire, elle renvoie 0 si la bibliothèque a été chargée. Si le nom de la bibliothèque ne comporte pas de nom de répertoire ou de "device" (DF0:, Work:, etc...), la recherche s'effectuera dans le chemin courant puis dans le chemin spécifié par le type d'outil LIBS.

Exemple : `LOADLIB("Symboles CMOS")`

Voir aussi : `REMLIB`, `LIBSPATH`.

## 1.227 Fonction ARexx LOADPREF

```
LOADPREF("nom_fichier")
```

Cette fonction a été supprimée dans la version 2.05.

## 1.228 Fonction ARexx LOCK

`LOCK`            Version 2.05 (argument supprimé)

Verrouille la fenêtre courante. C'est à dire que pendant toute l'exécution du script l'utilisateur ne peut plus avoir accès au document: les menus ne sont plus accessibles et les opérations faites à l'aide du clavier ou de la souris ne sont plus prises en compte. Si un verrouillage est effectué sur une fenêtre déjà bloquée, un compteur est incrémenté, il faut ensuite que la fenêtre soit débloquée autant de fois qu'elle a été bloquée avant qu'elle ne soit effectivement déverrouillée. Certaines fonctions, comme le choix d'un fichier, la sauvegarde, le chargement, l'impression, entraînent un verrouillage temporaire de toutes les fenêtres.

Valeur renvoyée: code supérieur ou égal à zéro si opération réussie, -1 si verrouillage impossible (manque de mémoire ?).

Nota : les fenêtres sont automatiquement débloquées lorsque l'exécution d'un script se termine. S'il se produit un blocage vous pouvez également mettre fin à cette situation en cliquant deux fois sur l'icône du programme située sur l'écran du Workbench. Cette fonction ne peut être appelée depuis une macro clavier (il y aurait risque de blocage total des entrées/sorties).

Voir aussi : `UNLOCK`.

## 1.229 Fonction ARexx MACRO



MACRO(x)

Cette fonction a été supprimée dans la version 2.04. Elle peut être remplacée par un appel aux fonctions EXEC(READMAP()).

## 1.230 Fonction ARexx MAP

MAP("combinaison touches", "séquence")

Cette fonction permet de programmer une macro-commande, qui sera exécutée lors de l'appui d'une certaine combinaison de touches. Les touches ALT, SHIFT et CTRL peuvent être utilisées pour définir la touche (il faut d'ailleurs utiliser au moins l'une d'entre elles, sauf pour les touches de fonction). Seules les touches définies par un seul caractère sont définissables (pour l'instant): vous ne pouvez définir par cette fonction les flèches ou certaines touches spéciales.

La séquence peut inclure n'importe quelle autre commande, elle doit être incluse entre des parenthèses (chaîne de caractères). La définition peut commencer par le signe =, le résultat de la macro alors automatiquement affiché, s'il y en a un.

À noter que ces définitions sont sauvées dans le fichier de configuration AmiCAD.keys lors d'une opération de sauvegarde des Préférences, et sont donc récupérées lors d'une opération de lecture ou au lancement du programme.

Vous pouvez également utiliser le script ARexx MapKey pour définir une séquence de touches.

Exemples :

```
MAP("shift-ctrl-a", "SAVE("Prog:Projets/Schémas/Essai schéma")")
MAP("CTRL-)", "CALL("Swap")")
MAP("F10", "WTOBACK")
```

Voir aussi :

READMAP, UNMAP, définition des touches de fonction.

## 1.231 Fonction ARexx MARK

MARK(numéro\_objet,...)

Permet de sélectionner un ou plusieurs objets. Les arguments doivent avoir une valeur comprise entre 1 et OBJECTS. Cette fonction renvoie le nombre d'objets qui ont été effectivement sélectionnés (les objets qui étaient déjà sélectionnés ne sont pas comptabilisés).

Voir aussi : UNMARK, MARKZONE.

---

## 1.232 Fonction ARexx MARKZONE

MARKZONE(x0, x1, y0, y1)

Permet de sélectionner les objets compris dans le rectangle défini par les points de coordonnées x0, y0 et x1, y1. Les éléments doivent être entièrement dans la zone pour être sélectionnés.

Cette fonction ne renvoie aucune valeur particulière.

Pour sélectionner l'ensemble des éléments du schéma, utilisez la commande suivante:

MARKZONE(0,0,WWIDTH-1,WHEIGHT-1)

Voir aussi : MARK, UNMARK.

## 1.233 Fonction ARexx MENU

MENU("intitulé menu")

Appelle le menu dont le titre (ou le début du titre seul) est spécifié, la recherche s'arrêtant pour le premier menu dont la chaîne correspond. Certains menus sont cependant inopérants pour cette fonction, reportez vous à la description des menus pour de plus amples précisions.

Il faut aussi savoir que les titres des menus comportant des espaces, comme "Effacer~fichier" comportent des "espaces solides", obtenus en frappant ALT-espace au clavier. Ceci a été rendu nécessaire par AmigaGuide qui ne retrouve pas les noeuds (nodes) dont le nom comporte des espaces (à moins que je n'aie fait une bêtise ?). Il faut donc bien taper le nom des menus concernés avec ces "espaces solides".

Ex: MENU ("Quit") Entraîne la fermeture de tous les tableaux

MENU ("Copier") Recopie des objets sélectionnés dans le tampon mémoire

Valeur renvoyée: nom du menu s'il a été trouvé. Une erreur est générée si ce n'est pas le cas (Valeur argument incorrecte).

Nota : l'intitulé du menu peut correspondre à la chaîne localisée ou à la chaîne originale, en anglais (Version 2.01). Il est préférable d'utiliser les chaînes en anglais afin que les scripts puissent fonctionner quelle que soit le langage utilisé.

Exemple : MENU("Copier") en français est équivalent à MENU("Copy").

## 1.234 Fonction ARexx MESSAGE

MESSAGE("chaîne")

Affiche une requête contenant le message spécifié. Ce texte peut contenir des sauts de lignes.

Exemple : MESSAGE("Ceci est un"+CHR(10)+"message.")

---

## 1.235 Fonction ARexx MESURE

MESURE(dimension fenêtre)

Renvoie une des dimensions de la fenêtre courante. La valeur qui est renvoyée dépend de celle passée en argument :

- égal 0: coordonnée gauche de l'emplacement de la fenêtre
- égal 1: coordonnée haute de l'emplacement de la fenêtre
- égal 2: largeur de la fenêtre (en pixels)
- égal 3: hauteur de la fenêtre (en pixels)
- égal 4: largeur maximale pouvant être prise par la fenêtre
- égal 5: hauteur maximale pouvant être prise par la fenêtre
- égal 6: largeur de l'écran (idem SWIDTH)
- égal 7: hauteur de l'écran (idem SHEIGHT)

Si la fenêtre est cachée la valeur renvoyée est toujours nulle, sauf pour les dimensions de l'écran, nulles uniquement si l'écran est fermé (quand toutes les fenêtres sont cachées).

Si la fenêtre est réduite la valeur déterminant sa dimension est renvoyée sous forme négative (soit pour les valeurs de l'argument égales de 0 à 3 compris).

Voir aussi : WINDOW

## 1.236 Fonction ARexx MODIF

MODIF Version 2.05 (argument supprimé)

Permet de savoir si la fenêtre courante contient un schéma ayant été modifié, sans avoir été sauvé.

La valeur renvoyée est nulle si le schéma n'a pas été modifié depuis la dernière sauvegarde, sinon elle est normalement égale à 1.

Exemple :

```
'MODIF'  
if result~=0 then 'MENU("Sauver")'
```

Autre écriture (plus rapide):

```
'IF(MODIF,MENU("Sauver"),0)'
```

Ce petit script peut être utilisé pour une sauvegarde automatique (voir script SauverAuto.AmiCAD).

## 1.237 Fonction ARexx MOVE

MOVE(numéro\_objet, dx, dy)

Cette fonction permet de déplacer un objet du nombre de pixels spécifié par les valeurs de dx et dy. S'il s'agit

---

d'un composant les fils qui lui sont connectés seront tirés, selon l'état du menu Tirer lignes.

Le numéro d'objet doit avoir sa valeur comprise entre 1 et OBJECTS. S'il est nul, tous les objets sélectionnés seront déplacés (Version 2.01).

Aucune valeur n'est renvoyée par cette fonction.

## 1.238 Fonction ARexx NBSHEET

NBSHEET(x)

Cette fonction a été supprimée dans la version 2.05 (voir NBSHEETS).

## 1.239 Fonction ARexx NBSHEETS

NBSHEETS("nom\_variable")

Version 2.05

Cette fonction permet de savoir combien il y a de fenêtres, et quels sont leurs port ARexx respectifs. Elle ne peut pas être utilisée en mode local. Le nom de la variable correspond au nom d'une variable ARexx qui contiendra les noms des ports de la façon suivante:

nom\_variable.0 contient le nom du port de la première fenêtre  
nom\_variable.1 contient le nom du port de la seconde fenêtre, etc  
Le nombre de fenêtres est contenu dans la variable result.

Exemple de script :

```
options results
'NBSHEETS("NOMS")'
n=result
do i=0 to n-1
  address value NOMS.i
  'MESSAGE("Document "+FILENAME("")+": port 'NOMS.i'")'
end
```

## 1.240 Fonction ARexx NEW

NEW("titre")

Cette fonction a été supprimée, utilisez la fonction NEWSHEET à la place.

## 1.241 Fonction ARexx NEWSHEET

NEWSHEET                      Version 2.05

Cette fonction ne prend aucun argument. Elle permet l'ouverture d'une nouvelle fenêtre. Elle renvoie un indice supérieur ou égal à 0 si l'opération a réussi, négatif si l'opération échoue. En fait l'indice correspond à celui de la nouvelle tâche qui a été créée. La fenêtre est ouverte après, à condition qu'il y ait assez de ressources système. Chaque nouvelle fenêtre possède son propre port ARexx, appelé AmiCAD.x où x correspond à l'indice renvoyé par cette fonction.

Les dimensions prises par la fenêtre sont celles spécifiées dans les préférences (dimensions document). Elles peuvent être changées en utilisant la fonction DIMSHEET.

Utilisez le script NewWindow si vous voulez ouvrir une nouvelle fenêtre, la valeur renvoyée peut en effet, dans certains cas ne pas concorder si la fenêtre ne peut être ouverte, ce script gère ces problèmes éventuels.

Exemple d'appel:

```
call 'NewWindow.AmiCAD' [nom_fichier]
code = result
if code < 0 then exit
address value "AmiCAD." || code
```

Ces quatre lignes permettent d'ouvrir une fenêtre qui prend le nom spécifié en argument (nom\_fichier). Le code retourné correspond à l'indice renvoyé par la fonction NEWSHEET.

## 1.242 Fonction ARexx NEXTSEL

NEXTSEL(numéro\_objet)

Renvoie le numéro d'objet sélectionné, suivant celui qui est spécifié en argument. Cette fonction permet ainsi, associée à FIRSTSEL, de parcourir tous les éléments sélectionnés.

## 1.243 Fonction ARexx OBJECTS

OBJECTS                      Version 2.05 (argument supprimé)

Cette fonction renvoie le nombre d'éléments appartenant à la fenêtre courante.

## 1.244 Fonction ARexx OPEN

OPEN("nom\_fichier#?")

Cette fonction permet de charger un fichier dans une nouvelle fenêtre, mais à la différence de la fonction LOAD, elle

---

permet également le chargement de plusieurs fichiers. Il suffit pour cela de spécifier des caractères génériques (#?[]) dans le nom du fichier, comme prévu sous DOS. Tous les schémas correspondant à ce format seront chargés. Une nouvelle fenêtre est ouverte pour chacun des schémas trouvés, correspondant à la demande.

La fonction renvoie le nombre de fenêtres ayant été ouvertes (v2.07).

Exemples :

OPEN("Travail:AmiCAD/Schémas/Projet\_TV/#?") demande le chargement de tous les fichiers du répertoire Projet\_TV

OPEN("#?.sch") demande de charger tous les fichiers ayant l'extension .sch, situés dans le répertoire courant

OPEN("Schémas/Carte ampli") demande le chargement du fichier "Carte ampli" situé dans le répertoire Schémas.

Voir aussi : CLOSE, LOAD.

## 1.245 Fonction ARexx OUTPUT

OUTPUT(nom\_connecteur, x, y)

Cette fonction permet de placer un connecteur de sortie à l'emplacement spécifié par les coordonnées x et y. La flèche est normalement dirigée vers la droite, sauf si vous utilisez la fonction SYMMETRY.

Si les valeurs de x et de y sont nulles, la chaîne spécifiée est placée sous le curseur, celui-ci est placé en mode placement de connecteur. L'utilisateur peut alors cliquer où il veut pour le placer (version 1.3).

Elle renvoie le numéro de l'objet qui a été placé, si elle réussit, sinon elle renvoie 0.

L'échelle verticale et l'échelle horizontale courantes sont prises en compte pour déterminer les dimensions de cet élément (voir SETSCALE, ROTATE, SYMMETRY).

Voir aussi : INPUT

## 1.246 Fonction ARexx PARTNAME

PARTNAME(numéro\_objet)

Cette fonction permet de savoir quel est le nom d'un composant. Elle renvoie une chaîne nulle si l'objet choisi n'est pas un

composant.

Le numéro d'objet doit avoir sa valeur comprise entre 1 et OBJECTS.

Voir aussi : PUTPART, LIBRARY.

## 1.247 Fonction ARexx PASTE

PASTE(*clip*, *x*, *y*)

Collage du contenu du tampon mémoire spécifié, à l'emplacement donné par les arguments *x* et *y*.

Aucune valeur particulière n'est renvoyée.

Voir aussi : COPY, CLIPUNIT.

## 1.248 Fonction ARexx PENWIDTH

PENWIDTH (*numéro\_objet*, *largeur\_trait*)

Permet de sélectionner la largeur du trait définissant le tracé de l'objet spécifié. La valeur du second paramètre doit être comprise entre -255 et 255 compris, sinon la fonction n'aura aucun effet.

Si le second argument est négatif, pour une ligne, celle-ci est alors du type spécial (ou personnalisé), s'il est nul le tracé sera effectué en pointillés.

Le numéro d'objet doit avoir sa valeur comprise entre 1 et OBJECTS.

Valeur renvoyée : largeur du trait utilisée AVANT l'exécution de la fonction.

## 1.249 Fonction ARexx PICKOBJ

PICKOBJ("message")

Affiche le message spécifié dans la barre de titre de la fenêtre, puis attend un clic de l'utilisateur dans la fenêtre. Renvoie le numéro de l'objet sur lequel a eu lieu le clic (0, s'il n'y en a pas). L'utilisateur peut faire défiler le texte à l'aide des ascenseurs et des boutons associés. Il peut aussi annuler l'opération en appuyant sur le bouton droit de la souris ou sur une touche du clavier, la valeur renvoyée est alors -1.

Voir aussi : GETPOINT.

---

## 1.250 Fonction ARexx PINCOL

PINCOL(numéro\_objet,broche)      Version 1.4

Cette fonction renvoie la colonne où se situe la broche d'indice spécifié appartenant à l'objet spécifié. Cet objet doit être un composant.

Attention l'indice de la broche ne correspond pas à son numéro, quand il existe, mais à l'ordre dans lequel les broches doivent apparaître dans une netlist de PSpice.

Voir aussi : DEVPINS, PINLINE, PINNUM.

## 1.251 Fonction ARexx PINLINE

PINLINE(numéro\_objet,broche)      Version 1.4

Cette fonction renvoie la ligne où se situe la broche d'indice spécifié appartenant à l'objet spécifié. Cet objet doit être un composant.

Attention l'indice de la broche ne correspond pas à son numéro, quand il existe, mais à l'ordre dans lequel les broches doivent apparaître dans une netlist de PSpice.

Voir aussi : DEVPINS, PINCOL, PINNUM.

## 1.252 Fonction ARexx PINNUM

PINNUM(numéro\_objet,broche)      Version 1.4

Cette fonction renvoie le numéro de la broche spécifiée. L'objet doit naturellement être un composant.

Attention l'indice de la broche correspond à l'ordre dans lequel les broches doivent apparaître dans une netlist de PSpice.

Voir aussi : DEVPINS, PINCOL, PINLINE.

## 1.253 Fonction ARexx PRINT

PRINT(rapport, rotation)

Imprime le schéma, en tenant compte des arguments  
rapport (facteur d'agrandissement, doit être compris entre 25 et 1000) et rotation (si égal à 0, le schéma est imprimé

---



comme il est visible à l'écran, si égal à 1, le schéma est imprimé avec une rotation de 90 degrés).

Renvoie 0 si tout se passe bien, un code d'erreur dans le cas contraire.

## 1.254 Fonction ARexx PUTPART

`PUTPART("nom_composant", x, y)`

Place le composant spécifié à l'emplacement donné par les arguments suivants. Attention la référence et la valeur ne sont pas placés, utilisez les fonctions SETREF et SETVAL ou LINKREF et LINKVAL pour cela.

L'échelle verticale et l'échelle horizontale courantes sont prises en compte pour déterminer les dimensions de cet élément (voir SETSCALE, ROTATE, SYMMETRY).

Cette fonction renvoie le code de l'objet qui a été placé si elle réussit, sinon elle renvoie 0.

Version 1.6:

Si les coordonnées x et y sont égales à -1, le composant spécifié devient le composant courant, c'est à dire celui qui sera placé lors du prochain appel au menu Dessin/Placer composant. Cette fonction remplace l'ancienne fonction GETPART.

Voir aussi : PARTNAME

## 1.255 Fonction ARexx READCONV

`READCONV(numéro_objet)`

Cette fonction permet de savoir quel est le type de symbole utilisé pour afficher un composant ou une jonction. Elle renvoie -1 si l'objet choisi est d'un autre type, 0 si cet élément est affiché normalement ou 1 si c'est le symbole alterné qui est utilisé.

Le numéro d'objet doit avoir sa valeur comprise entre 1 et OBJECTS, s'il est nul c'est la valeur du mode courant qui est renvoyée.

Voir aussi : CONVERT, menu Dessin/Alterné~symbole

## 1.256 Fonction ARexx READDEF

---

```
READDEF("fonction")
```

Cette fonction permet de connaître la définition associée à une fonction. Le nom de la fonction passée en argument doit figurer en MAJUSCULES, entre parenthèses. Cette fonction doit bien sûr avoir été préalablement définie à l'aide de la fonction DEF, sinon une chaîne nulle est renvoyée. Vous ne pouvez lire les définitions préprogrammées, comme ABS, ARC, ASC, etc...

Exemple :

```
READDEF("CIRCLE")    renvoie la définition associée à la  
fonction CIRCLE, si elle existe.
```

Voir aussi : DEF

## 1.257 Fonction ARexx READDEV

```
READDEV(numéro_objet)
```

Cette fonction permet de savoir quel est la porte ou le circuit utilisé par un composant. Elle renvoie -1 si l'objet choisi n'est pas un composant, 0 si ce composant ne comprend qu'un circuit ou bien le numéro du circuit sélectionné (alors supérieur ou égal à 1).

Le numéro d'objet doit avoir sa valeur comprise entre 1 et OBJECTS.

Voir aussi : SETDEV, GETDEVS.

## 1.258 Fonction ARexx READMAP

```
READMAP("combinaison touches")
```

Cette fonction renvoie une chaîne de caractère contenant la séquence de commande associée à la combinaison de touches spécifiée. Elle permet ainsi de créer des macros qui peuvent s'appeler les unes les autres. Sa vocation est cependant de vérifier qu'une séquence n'existe pas déjà pour une combinaison de touches donnée, ou de déterminer connaître la macro déjà programmée afin de l'éditer (comme dans le script MapKey).

Exemples :

```
READMAP("shift-ctrl-a")  
READMAP("CTRL-i")  
READMAP("ALT-$\mathrm{\mu}$")  
EXEC(READMAP("F1"))    Exécute la commande associée à la touche F1
```

Voir aussi : MAP, UNMAP

---

## 1.259 Fonction ARexx READTEXT

`READTEXT(numéro_objet)`

Renvoie le texte associé à un objet, cet objet peut être un élément de texte, ou un connecteur (entrée ou sortie). Si ce texte n'existe pas ou si l'objet est d'un type différent la fonction renvoie une chaîne nulle.

Le numéro d'objet doit avoir sa valeur comprise entre 1 et OBJECTS.

Voir aussi : SETTEXT

## 1.260 Fonction ARexx REMLIB

`REMLIB("nom_bibliothèque")`

Supprime la bibliothèque spécifiée de la mémoire. Attention : aucun élément ne doit être présent sur un document ou l'opération échouera. Cette opération peut permettre de gagner de la mémoire, en supprimant par exemple une bibliothèque qui n'est plus utilisée.

Vous pouvez aussi utiliser le bouton "Supprimer" de la requête de choix d'un composant.

Exemple : `REMLIB("Symboles TTL")`

Voir aussi : LOADLIB

## 1.261 Fonction ARexx REQFILE

`REQFILE("titre_requête", "chemin", "fichier")`

Ouvre la boîte de requête de fichier, avec le titre spécifié. La requête s'ouvre en explorant le chemin spécifié. Ce chemin doit correspondre à celui d'un répertoire ou d'un volume.

Renvoie le nom du fichier sélectionné ou une chaîne nulle si l'utilisateur clique sur le bouton Annule.

Note: le troisième argument est apparu à partir de la version 2.0, il permet de spécifier un nom de fichier initial dans la requête.

Exemple :

`REQFILE("Choisissez un fichier", "Work:AmiCAD", "Tracé_"+FILEPART(""))`

---

## 1.262 Fonction ARexx REQLIST

REQLIST (titre, entréel, ...) Version 2.04

Cette fonction permet l'ouverture d'une fenêtre possédant le titre spécifié. Elle contient autant de lignes déterminées par les arguments qui suivent (entréel, etc...). Elle renvoie le numéro de l'entrée (1 à n) sur lequel l'utilisateur clique avec le bouton gauche de la souris, ou -1 si l'opération est annulée (clic sur le bouton de fermeture de la fenêtre).

Si la fenêtre n'a pu être ouverte (manque mémoire ou autre), la valeur 0 est renvoyée.

## 1.263 Fonction ARexx REQSHEET

REQSHEET("titre") Version 2.05 (modifiée)

Cette fonction permet de choisir, à l'aide d'une boîte de requête, un schéma parmi ceux qui sont chargés en mémoire. Le titre est affiché en haut de la requête.

Le nom du port ARexx correspondant au schéma choisi est renvoyé. Si l'utilisateur ferme la fenêtre sans choisir un document, ou bien si un problème survient une chaîne nulle ("" ) est renvoyée.

## 1.264 Fonction ARexx REQUEST

REQUEST("titre")

Affiche une requête avec le texte spécifié (jusqu'à treize lignes de texte, séparés par des sauts de ligne), et deux boutons OUI et NON. Si l'utilisateur clique sur le bouton OUI ou appuie sur ENTRÉE, cette fonction renvoie 1.

Si l'utilisateur clique sur le bouton NON ou appuie sur la touche ESC, la valeur renvoyée est 0.

En cas de problème (manque mémoire par exemple), la valeur renvoyée est négative.

## 1.265 Fonction ARexx RESET

RESET(variable,...)

Cette fonction est identique à la fonction INIT, cependant le type des variables n'est pas réinitialisé. Si la variable est du type numérique elle prend la valeur 0, si c'est une chaîne de caractères elle devient une chaîne vide ("" ).

---

Le nombre d'arguments est quelconque.

Exemple : RESET(A,B,C)

Voir aussi : INIT

## 1.266 Fonction ARexx REXXPORT

REXXPORT("nom\_fichier")

Version 2.05

Cette fonction permet de connaître le nom du port ARexx associé à un document, s'il est présent en mémoire. Une chaîne nulle est renvoyée si le document n'existe pas.

L'argument n'a pas à inclure le chemin complet (optionnel). Si c'est une chaîne nulle ("") le nom du port ARexx de la fenêtre courante est renvoyé.

Voir aussi : NBSHEETS

## 1.267 Fonction ARexx ROTATE

ROTATE(numéro\_objet, rotations)

Fait tourner un objet du nombre de quarts de tour spécifié. Si le numéro d'objet est nul, c'est le mode de placement courant qui est affecté: les placements effectués ensuite tiendront compte de cette nouvelle position, que ce soit par une macro (PUTPART par exemple) ou un menu.

Cette fonction renvoie 0 si l'opération a échoué (par manque de place par exemple), sinon elle renvoie 1.

Voir aussi : SYMMETRY, SETSCALE, GETPOS.

## 1.268 Fonction ARexx SAVE

SAVE("nom\_fichier")

Sauve le document sous le nom spécifié. Si le fichier existait déjà aucun avertissement n'a lieu. Renvoie 1 si aucun problème ne survient, sinon 0. Un fichier .info comportant une icône est créée si le menu Préférences/Sauver~icône est marqué, un fichier comportant l'extension .bis est créé si le menu Préférences/Copie schéma sauvé est marqué.

Voir aussi: SAVECOPY, SAVEICON.

---

## 1.269 Fonction ARexx SAVEALL

SAVEALL      Version 2.05 (argument supprimé)

Cette fonction permet de sauver le contenu du document en mémoire. Son intérêt est de pouvoir annuler les modifications induites par un script ARexx à l'aide du menu "Édition/Restaurer". L'appel à cette fonction doit naturellement être faite au début au script.

## 1.270 Fonction ARexx SAVECLIP

SAVECLIP (unité, "nom\_fichier")

Cette fonction permet de sauver le clip spécifié dans le fichier spécifié. Naturellement le clip doit auparavant avoir été rempli à l'aide de la fonction COPY, par exemple.

Voir aussi : LOADCLIP, CLIPUNIT.

## 1.271 Fonction ARexx SAVECOPY

SAVECOPY(1/0/-1)

Cette fonction permet de déterminer, comme le menu Préférences/Copie schéma ↔ sauvé, si une copie du schéma sera créée lors d'une opération de sauvegarde. Cette fonction renvoie 1 si le menu était marqué AVANT l'exécution de la fonction, ou 0 s'il ne l'était pas. Si l'argument vaut -1 l'état du menu est changé.

Pour lire l'état du menu sans le modifier, utilisez la commande SAVECOPY(-2).

Voir aussi : SAVEICON.

## 1.272 Fonction ARexx SAVEICON

SAVEICON(1/0/-1)

Cette fonction permet de fixer, comme le menu Préférences/Sauver~icône, si une icône sera créée lors d'une opération de sauvegarde.

Cette fonction renvoie 1 si le menu était marqué AVANT l'exécution de la fonction, ou 0 s'il ne l'était pas. Si l'argument vaut -1 l'état du menu est changé.

Pour lire l'état du menu sans le modifier, utilisez la commande SAVEICON(-2).

Voir aussi : SAVECOPY.

---

## 1.273 Fonction ARexx SAVEIFF

```
SAVEIFF("nom_fichier",rapport_affichage)
```

Sauve le document au format ILBM (image BitMap) en noir et blanc. Renvoie 0 si tout s'est bien passé.

Version 2.01: le rapport d'affichage correspond à un facteur d'agrandissement (ou de diminution) de la taille de l'image, il est exprimé en pourcentage (égal à 100 pour avoir une taille d'image normale).

Exemple :

```
SAVEIFF("Work:Images/Schéma_amplificateur", 200)
```

Voir aussi : menu Projet/Sauver format IFF

## 1.274 Fonction ARexx SAVEKEYS

```
SAVEKEYS("nom fichier")
```

Cette fonction a été supprimée dans la version 2.05.

## 1.275 Fonction ARexx SAVEPREF

```
SAVEPREF("nom fichier")
```

Cette fonction a été supprimée dans la version 2.05.

## 1.276 Fonction ARexx SCREEN

```
SCREEN (mode, largeur, hauteur,"fonte",taille)
```

Cette fonction permet de choisir la résolution de l'écran d'AmiCAD sans avoir à passer par la requête Asl, comme avec le menu "Préférences/Choix mode écran". De plus vous pouvez spécifier quelle sera la fonte utilisée pour les menus, les boîtes de requêtes, etc... Le nom de la fonte doit être complet, y compris l'extension ".font"

Cependant il vous faudra connaître les valeurs associées aux divers types d'écran, un moyen simple pour les connaître est de l'afficher avec la fonction SCRMODE, en tapant la commande =SCRMODE dans la requête suivant l'appel au menu Macros/Mode direct.

Ainsi les valeurs suivantes sont possibles:

- 561188 pour un écran en super résolution entrelacée (SUPER72)
  - 561152 pour un écran en SUPER72 haute résolution.
  - 233509 pour un écran MultiScan Productivité entrelacée
  - 167936 pour un écran PAL haute résolution
  - 135168 pour un écran PAL basse résolution
-

La fonction renvoie le code en vigueur AVANT son exécution.

L'écran comprend toujours seize couleurs.

Exemples :

SCREEN(561188,800,600,"courier.font",15) ' écran "normal" (Super 72)

SCREEN(135168,320,200,"topaz.font",8) ' écran basse résolution (pour " ←  
zoomer")

Voir aussi : SCRMODE, SHEIGHT, SWIDTH, FONTNAME, FONTSIZE.

## 1.277 Fonction ARexx SCRMODE

SCRMODE

Cette fonction renvoie la valeur associée au mode d'écran courant.  
Elle n'a besoin d'aucun argument.

Voir aussi : SCREEN

## 1.278 Fonction ARexx SECURITY

SECURITY(nombre boucles)

Cette fonction détermine le nombre maximal de boucles pouvant être effectuées par une des fonctions FOR ou WHILE. Ceci permet de sortir des boucles sans fin assez simplement. La valeur par défaut est égale à 500. Vous pouvez entrer une valeur allant de 1 jusqu'à  $2^{31}-1$  (2147483647). La fonction renvoie la valeur en cours avant son exécution.

Vous pouvez donner à cette fonction un argument de grande valeur sans craindre de blocage, il est effet maintenant possible d'interrompre une boucle en appuyant simultanément sur les touches CTRL, ALT et ESC.

Si la valeur passée en argument est nulle seule la valeur actuelle est renvoyée, sans modification.

## 1.279 Fonction ARexx SELECT

SELECT("texte")

Cette fonction permet d'ouvrir une boîte de requête comportant un nombre variable de boutons, comportant chacun un texte choisi par l'utilisateur. Si l'utilisateur clique sur l'un de ces boutons, la fonction renvoie le rang du bouton (en commençant par la valeur 1 pour le premier, 2 pour le second, etc...).

---



Le format du texte passé en argument doit être le suivant:

- une première ligne, affichée en titre,
- une seconde ligne, correspondant au texte du premier bouton
- une troisième ligne, correspondant au texte du second bouton,
- et ainsi de suite, autant de lignes que de boutons...

Chacune des lignes est séparée des suivantes par un saut de ligne (CHR(10)).

Si la valeur renvoyée est négative ou nulle, c'est que l'utilisateur a appuyé sur le bouton droit ou sur la touche Esc, ou bien que la requête n'a pu être ouverte.

Exemple :

```
SELECT("Nombre de circuits ?"+CHR(10)+"10 circuits"+CHR(10)+"20 circuits"+CHR(10)+"À déterminer")
```

## 1.280 Fonction ARexx SELFILE

```
SELFILE("nom_fichier")
```

Cette fonction a été supprimée dans la version 2.05.

Voir aussi : REXXPORT

## 1.281 Fonction ARexx SELSHEET

```
SELSHEET(fenêtre)
```

Cette fonction a été supprimée dans la version 2.05.

## 1.282 Fonction ARexx SETARRAY

```
SETARRAY(tableau, indice, valeur)
```

Cette fonction permet d'affecter une valeur à une variable appartenant à un tableau. L'indice doit être compris entre 0 et la valeur maximale déterminant la taille du tableau (diminuée de 1), si cet indice est inférieur à 0, tous les éléments du tableau prennent la valeur spécifiée. La valeur doit toujours être du même type (nombre ou chaîne de caractères, la première affectation détermine le type utilisé par la suite).

Exemple:

```
'DIMARRAY(N,100)'
'FOR(I=0,I<100,SETARRAY(N,I,I*2),I=I+1)'
'GETARRAY(N,50)'           retourne 100
'SETARRAY(N,-1,0)'
```

'GETARRAY(N,50)'                      retourne 0

Voir aussi: ARRAYDIM, DIMARRAY, DELARRAY, GETARRAY

## 1.283 Fonction ARexx SETCOLOR

SETCOLOR(couleur,niveau\_rouge,niveau\_vert,niveau\_bleu)

Cette fonction permet de choisir une couleur de l'écran AmiCAD. Le nombre correspondant à cette couleur doit être compris entre 0 et 15 compris (l'écran a seize couleurs).

Les niveaux de couleurs sont des mots long, de 32 bits, pour spécifier du blanc, passez trois arguments égaux à 0xFFFFFFFF, pour ramener un niveau de couleur à un niveau intermédiaire, passez un argument de valeur 0x7FFFFFFF.

Aucune valeur n'est renvoyée par cette fonction.

Voir aussi : GETCOLOR.

## 1.284 Fonction ARexx SETDEV

SETDEV(numéro\_objet, numéro\_circuit)

Cette fonction permet de choisir le numéro de circuit ou de porte pour un composant qui en comporte plusieurs. Elle est utile pour les circuits comportant de multiples portes comme les circuits TTL 7400 par exemple.

Le numéro d'objet doit avoir sa valeur comprise entre 1 et OBJECTS, s'il est nul c'est le mode de placement courant qui est affecté, c'est à dire que le prochain composant qui sera placé aura ce numéro de circuit (s'il est correct, c'est à dire pas trop grand).

La fonction peut renvoyer trois résultats possibles:

- 0 si l'objet spécifié n'est pas un composant,
- 1 si l'opération a réussi,
- -1 si le numéro de circuit est incorrect.

Voir aussi : READDEV, GETDEVS.

## 1.285 Fonction ARexx SETFILL

SETFILL(0/1/-1)

Détermine quel est le mode de remplissage utilisé pour dessiner les objets du type rectangle, triangle ou ellipse (v2.0).

---

Si l'argument vaut 1, les objets seront pleins, s'il vaut 0 il seront vides. S'il vaut -1 le mode actuel reste inchangé, la valeur en vigueur avant l'appel de la fonction est alors renvoyée.

## 1.286 Fonction ARexx SETGRID

SETGRID(pas\_grille)

Détermine la taille de la grille utilisée pour placer les composants sur le schéma. À noter que cette fonction ne met pas à jour la grille à l'écran, ce qui peut poser des problèmes, utilisez le menu "Dessin/Redessiner tout" si vous souhaitez qu'elle soit mise à jour, en utilisant par exemple la commande suivante :

SETGRID(15):MENU("Redessiner")

Valeur renvoyée : valeur du pas utilisée avant l'exécution de la commande.

Utilisez la commande SETGRID(0) pour connaître le pas de la grille sans le modifier.

Voir aussi : SNAPGRID.

## 1.287 Fonction ARexx SETLABEL

SETLABEL(numéro\_objet,"Label")

La valeur spécifiée (sous forme de chaîne de caractères) est attribuée à l'objet spécifié. Si l'objet spécifié possédait déjà un label, il est remplacé. Cet objet doit naturellement être un fil.

Le numéro de l'objet ajouté ou modifié est renvoyé si l'opération a réussi (0 dans le cas contraire). Ce numéro est identique à celui renvoyé par la fonction GETLABEL (Version 2.08).

Exemple : SETLABEL(FIRSTSEL,"Label1")

Voir aussi : LINKLAB, UNLINK, GETLABEL

## 1.288 Fonction ARexx SETORTHO

SETORTHO(action)

Cette fonction permet de déterminer si les tracés seront ou

non effectués à 45\textdegree{} lors de leur placement par l'utilisateur. Elle est analogue au menu Dessin/Tracé orthogonal.

Si action est égal à 1, le menu sera marqué,  
si action est égal à 0, le menu est démarqué,  
si action vaut -1, le menu change d'état.  
Pour toute autre valeur il ne se passe rien.

La fonction renvoie l'état du menu (1 quand il est marqué, 0 sinon) tel qu'il était avant son exécution.

## 1.289 Fonction ARexx SETPINS

SETPINS (numéro\_objet, ON/OFF)

Cette fonction permet de déterminer si les numéros de broches d'un composant seront affichés ou non. Le numéro d'objet doit être compris entre 1 et OBJECTS, s'il est nul la commande s'applique au mode de placement courant des composants.

Si le second argument est supérieur à 0, l'affichage est validé, s'il est égal à 0, l'affichage est supprimé, s'il est inférieur à 0 rien n'est changé (utilisé pour lire l'état du placement).

La valeur renvoyée est égale à 1 si l'affichage était validé AVANT l'application de la fonction, à 0 dans le cas contraire.  
Si l'objet spécifié n'est pas un composant, la valeur renvoyée est toujours égale à -1.

Exemple : SETPINS(FIRSTSEL,-1) renvoie l'état du 1er objet sélectionné

Voir aussi : Menu Dessin/Placer~numéros~bornes

## 1.290 Fonction ARexx SETREF

SETREF (numéro\_objet, "référence")

La valeur spécifiée (sous forme de chaîne de caractères) est attribuée à l'objet spécifié. Si l'objet spécifié possédait déjà une référence, elle est remplacée. Cet objet doit naturellement être un composant.

Le numéro de l'objet ajouté ou modifié est renvoyé si l'opération a réussi (0 dans le cas contraire). Ce numéro est identique à celui renvoyé par la fonction GETREF (Version 2.08).

Exemple : SETREF(FIRSTSEL, "R2")

Voir aussi : LINKREF, SETVAL, LINKVAL

## 1.291 Fonction ARexx SETSCALE

SETSCALE(numéro\_objet, échelle\_horizontale, échelle\_verticale)

Le numéro d'objet doit avoir sa valeur comprise entre 1 et OBJECTS. S'il est nul, la fonction fixe les valeurs par défaut des échelles, utilisées lors du placement des composants, comme avec le menu Dessin/Placer composant. Ces valeurs peuvent aussi être déterminées à l'aide des menus Préférences/Échelle verticale et Préférences/Échelle horizontale. Si l'une des valeurs des arguments correspondant à une échelle est négative, cette échelle restera inchangée.

Cette fonction renvoie 1 si elle a réussi, 0 dans le cas contraire (manque de place, échelle incorrecte...).

Si un script est appelé à changer l'échelle de placement courante, aussi bien horizontale que verticale, il est souvent souhaitable de rétablir les valeurs qui étaient en vigueur avant l'exécution du script. Pour cela la structure suivante peut être utilisée :

```
/* Soit à utiliser l'échelle horizontale x et l'échelle verticale y */
'_EH_=HSCALE(0):_EV_=VSCALE(0):SETSCALE(0,x,y)'
...
'SETSCALE(0,_EH_,_EV_)'
```

Voir aussi : HSCALE, VSCALE

## 1.292 Fonction ARexx SETTEXT

SETTEXT(numéro\_objet, "chaîne")

Cette fonction permet de fixer le texte associé à un objet. Cet objet doit être du type texte, valeur ou référence de composant, ou bien un connecteur (entrée ou sortie).

Le numéro d'objet doit avoir sa valeur comprise entre 1 et OBJECTS.

La valeur renvoyée est égale à 1 si la fonction a réussi, sinon elle est nulle.

Voir aussi : READTEXT

## 1.293 Fonction ARexx SETVAL

SETVAL(numéro\_objet, "valeur")

La valeur spécifiée (sous forme de chaîne de caractères) est attribuée à l'objet spécifié. Si l'objet spécifié possédait

---

déjà une valeur, elle est remplacée. Cet objet doit naturellement être du type composant.

Le numéro de l'objet ajouté ou modifié est renvoyé si l'opération a réussi (0 dans le cas contraire). Ce numéro est identique à celui renvoyé par la fonction GETVAL (Version 2.08).

Exemple : SETVAL(FIRSTSEL,"100k")

Voir aussi : LINKVAL, SETREF, LINKREF

## 1.294 Fonction ARexx SGN

SGN(nombre)

Renvoie 1 si l'argument est positif, -1 s'il est négatif, zéro s'il est nul.

## 1.295 Fonction ARexx SHEIGHT

SHEIGHT

Cette fonction renvoie la hauteur de l'écran courant. Elle ne nécessite aucun argument.

Voir aussi: SCREEN, SWIDTH, SCRMODE

## 1.296 Fonction ARexx SNAPGRID

SNAPGRID(action)

Cette fonction permet de déterminer si les éléments seront ou non alignés sur la grille lors de leur placement par l'utilisateur. Elle est liée au menu Dessin/Aligner~sur~grille.

Si action est égal à 1, le menu sera marqué,  
si action est égal à 0, le menu est démarqué,  
si action vaut -1, le menu change d'état.  
Pour toute autre valeur il ne se passe rien.

La fonction renvoie l'état du menu (1 quand il est marqué, 0 sinon) tel qu'il était avant son exécution.

Voir aussi: SETGRID.

---

## 1.297 Fonction ARexx STOBACK

STOBACK

Cette fonction fait passer l'écran d'AmiCAD en arrière-plan.  
Elle ne nécessite aucun argument.

Voir aussi : STOFRONT WTOBACK WTOFRONT

## 1.298 Fonction ARexx STOFRONT

STOFRONT

Cette fonction fait passer l'écran d'AmiCAD au premier plan.  
Elle ne nécessite aucun argument.

Voir aussi : STOBACK WTOBACK WTOFRONT

## 1.299 Fonction ARexx STR

STR(nombre)

Renvoie la chaîne de caractères correspondant à un nombre.  
La base utilisée pour la conversion est la base 10 (décimal).

## 1.300 Fonction ARexx SWIDTH

SWIDTH

Cette fonction renvoie la largeur de l'écran courant.  
Elle ne nécessite aucun argument.

Voir aussi : SCREEN, SHEIGHT, SCRMODE

## 1.301 Fonction ARexx SYMMETRY

SYMMETRY(numéro\_objet, position)

Cette fonction permet de symétriser ou non un objet, par rapport à son axe vertical (ou son axe horizontal, s'il a été tourné d'un ou trois quarts de tour).

Le numéro d'objet doit avoir sa valeur comprise entre 1 et OBJECTS. Si ce numéro est nul, la fonction modifie la façon de placer un composant (menu "Dessin/Placer composant").

L'argument position peut prendre trois valeurs :

- égal à 1: la symétrie est réalisée,
- égal à 0: pas de symétrie,
- égal à -1: inversion de la position.

La valeur renvoyée est égale à 1 s'il y a eu une modification de la position de l'objet, 0 dans le cas contraire.

Voir aussi : ROTATE, SETSCALE, GETPOS.

### 1.302 Fonction ARexx TEST

TEST(numéro\_objet)

Permet de savoir si un objet est sélectionné ou non.  
Le numéro d'objet doit avoir sa valeur comprise entre 1 et OBJECTS.

La fonction renvoie 1 si l'objet est sélectionné, 0 dans le cas contraire.

### 1.303 Fonction ARexx TIME

TIME(secondes)

Renvoie l'heure courante, si l'argument est non nul, les secondes sont comprises.

Exemples :

TIME(0)	renvoie 23:24
TIME(1)	renvoie 23:24:27
WRITE(TIME(1),0,0)	écrit l'heure courante "sous" le curseur

Voir aussi : DATE

### 1.304 Fonction ARexx TITLE

TITLE("titre")

Spécifie le titre de la fenêtre courante. Si l'argument est une chaîne nulle, le titre reprend son état normal, c'est à dire que le nom du fichier y réapparaît.  
Cette fonction permet de signaler une opération un peu longue en prévenant l'utilisateur, sans pour autant bloquer la fenêtre ni le programme.

Exemple de script :

```
'TITLE("Traitement en cours...")'  
... traitement...  
'TITLE("")'           on remet le titre normal
```



### 1.305 Fonction ARexx TOOLBAR

```
TOOLBAR("fichier")
```

Charge la barre d'outils spécifiée. Le fichier doit être au format IFF, l'icône doit comporter les tooltypes nécessaires:

- COLUMNS: nombre d'icônes par ligne,
- LINES: nombre de lignes d'icônes,
- COMMAND(x,y): commande associée à l'icône située à l'emplacement spécifié (une commande par icône).
- XBAR: abscisse placement fenêtre (optionnel),
- YBAR: ordonnée placement fenêtre (optionnel).

Cette fonction utilise la bibliothèque iff.library pour le chargement des images.

Pour créer une nouvelle barre d'outils, créez une image à l'aide d'un programme de dessin (comme PPaint, DPaint ou tout autre), sauvez cette image en la sauvant en tant que brosse (brush). Découpez cette brosse en gardant un pixel au moins sur les côtés, attention: les icônes doivent toutes avoir les mêmes dimensions. Éditez ensuite les tooltypes de l'image et ajoutez-y d'abord les outils COLUMNS et LINES au moins. Les commandes pourront être éditées à l'aide de AmiCAD en sélectionnant la barre (il faut cliquer sur la barre de déplacement de la fenêtre) puis en cliquant à l'aide du bouton droit de la souris sur l'icône à modifier. Un menu contextuel est alors ouvert, choisissez l'action à effectuer (éditer, copier, coller, décaler les commandes ou sauver les commandes dans l'icône). N'oubliez pas de sauver les commandes quand tout fonctionne correctement.

Attention: si vous utilisez un programme de dessin pour modifier un fichier image, celui-ci écrase quelque fois les tooltypes de cette image lors de l'opération de sauvegarde, ne travaillez alors que sur une copie de l'image.

Cette fonction renvoie un résultat nul si le chargement s'est bien passé (version 2.05).

### 1.306 Fonction ARexx TRIANGLE

```
TRIANGLE(x0,y0,x1,y1,x2,y2)
```

Trace un triangle selon les coordonnées spécifiées.

La largeur du tracé dépend du mode courant (voir DRAWMODE) et du mode de remplissage (voir SETFILL).

Le numéro de l'objet qui a été créé est renvoyé (0 si erreur).

Utilisez la fonction COORDS pour connaître les coordonnées d'un triangle existant.

---

### 1.307 Fonction ARexx TXHEIGHT

```
TXHEIGHT("texte")
```

Renvoie la hauteur du texte spécifié, en pixels. Cette fonction prend en compte l'échelle courante, ainsi que le mode de placement (rotation éventuelle). Dans tous les cas c'est l'espace vertical occupé qui est renvoyé.

Voir aussi : TXWIDTH

### 1.308 Fonction ARexx TXWIDTH

```
TXWIDTH("texte")
```

Renvoie la largeur du texte spécifié, en pixels. Cette fonction prend en compte l'échelle courante, ainsi que le mode de placement (rotation éventuelle). Dans tous les cas c'est l'espace horizontal occupé qui est renvoyé.

Cette fonction est utile pour centrer un texte dans un rectangle.

Exemple :

```
options results
texte="Essai texte"
xx=100;y=100
'TXWIDTH("'texte'")'
l=result
'TXHEIGHT("'texte'")'
h=result
'DRAW('xx','y','xx+l','y'):DRAW('xx+l','y','xx+l','y+h')'
'DRAW('xx+l','y+h','xx','y+h'):DRAW('xx','y+h','xx','y')'
'WRITE("'texte'",'xx','y+h')
```

Voir aussi : TXHEIGHT

### 1.309 Fonction ARexx TYPE

```
TYPE(numéro_objet)
```

Cette fonction permet de connaître la nature d'un objet. Le numéro d'objet doit avoir sa valeur comprise entre 1 et OBJECTS.

La valeur renvoyée dépend de son type :

- égale à 1: c'est un composant
  - égale à 2: c'est un fil de liaison normal
  - égale à 3: c'est un arc de cercle
  - égale à 4: c'est un texte
  - égale à 5: c'est une référence de composant
  - égale à 6: c'est une valeur de composant (ou son type)
  - égale à 7: c'est une connexion
-

- égale à 8: c'est un trait en pointillés
- égale à 9: c'est un bus
- égale à 10: c'est une ellipse
- égale à 11: c'est un connecteur d'entrée
- égale à 12: c'est un connecteur de sortie
- égale à 15: c'est un trait double
- égale à 21: c'est une ligne personnalisée (largeur quelconque)
- égale à 22: c'est un rectangle
- égale à 24: c'est un rectangle plein
- égale à 26: c'est un triangle
- égale à 27: c'est un triangle plein
- égale à 28: c'est un label (Version 2.08).

Vous pouvez ainsi définir des fonctions permettant de reconnaître les objets :

```
DEF COMPOSANT(O) = IF (TYPE(O) == 1, 1, 0)
DEF FIL(O) = IF (TYPE(O) == 2, 1, 0)
DEF CONNECTEUR(O) = IF ((TYPE(O)==11) | (TYPE(O)==12), 1, 0)
DEF LIGNE(O) = IF ((TYPE(O)==2) | (TYPE(O)==8) | (TYPE(O)==9) | (TYPE(O)==15) | (TYPE(O)==21), 1, 0) ←
```

Chacune de ces fonctions renvoie 1 si l'objet est du type testé sinon elles renvoient 0.

Nota: le script InitObjets.AmiCAD initialise des variables au nom correspondant à la valeur du type.

Ainsi on y trouve COMPOSANT=1, ceci permet alors d'utiliser le nom d'un objet et non la valeur du type qui lui associée. Pour utiliser ces définitions il suffit d'appeler ce script au début d'un script qui l'utilise (voir le script SaveNets.AmiCAD pour exemple).

### 1.310 Fonction ARexx UNGROUP

UNGROUP(groupe)

Version 1.4

Le groupe dont le numéro est spécifié est dissocié.  
Renvoie le nombre d'éléments appartenant au groupe.

Voir aussi : GROUP.

### 1.311 Fonction ARexx UNLINK

UNLINK(numéro\_objet)

Le numéro d'objet doit avoir sa valeur comprise entre 1 et OBJECTS.

Cette fonction permet de briser les liens existant entre un composant et sa référence et (ou) sa valeur. Si c'est le numéro du composant qui est passé en argument, les deux liens sont supprimés, s'ils existent. Si c'est le numéro de la valeur (ou de la référence) qui est passé, seul cet élément

est séparé. Le nombre de liens ayant été supprimés est renvoyé. Si l'objet spécifié est d'un autre type, la valeur renvoyée est égale à -1.

Version 2.08: cette fonction permet aussi de séparer un fil et son label. Le numéro de l'objet spécifié peut alors être celui du fil ou de son label.

Exemples :

```
UNLINK(FIRSTSEL)
UNLINK(PICKOBJ("Cliquez sur un composant"))
```

Voir aussi : LINKVAL, LINKREF, LINKLAB

## 1.312 Fonction ARexx UNLOCK

UNLOCK            Version 2.05 (argument supprimé)

Annule le verrouillage de la fenêtre courante, réalisé par la fonction LOCK.

## 1.313 Fonction ARexx UNMAP

```
UNMAP("combinaison touches")
```

Cette fonction permet d'annuler la programmation d'une macro-commande, associée à une certaine combinaison de touches. Les touches ALT, SHIFT et CTRL peuvent être utilisées pour définir la touche (il faut d'ailleurs utiliser au moins l'une d'entre elles, sauf pour les touches de fonction).

Exemples :

```
UNMAP("shift-ctrl-a")
UNMAP("ctrl-shift-a")
UNMAP("CTRL-")
UNMAP("ALT- $\mu$ ")
UNMAP("F2")
```

Voir aussi : MAP, READMAP

## 1.314 Fonction ARexx UNMARK

```
UNMARK(numéro_objet,...)
```

Annule le marquage des objets spécifiés.

Les numéros d'objets doivent avoir leurs valeurs comprises entre 1 et OBJECTS.

Renvoie le nombre de sélections ayant été annulées.

---

Note : l'appel de la fonction avec un argument égal à -1 entraîne l'annulation de la sélection de tous les objets du document courant (renvoie alors 0).

Voir aussi : MARK, MARKZONE

### 1.315 Fonction ARexx VAL

VAL("chaîne")

Renvoie la valeur numérique correspondant à la chaîne de caractères passée en argument. Seuls les caractères correspondant à des nombres entiers sont pris en compte. Le nombre doit figurer sous forme décimale, sauf s'il est précédé du préfixe \$, qui signale qu'il s'agit d'un nombre sous forme hexadécimale.

Exemples :

VAL("14")	renvoie 14
VAL("14.3")	renvoie 14
VAL("\$5F")	renvoie 95 (le \$ spécifie une chaîne hexadécimale)

Voir aussi : STR

### 1.316 Fonction ARexx VERSION

VERSION(type)

Cette fonction renvoie le numéro de version du programme quelle que soit la valeur de l'argument (version 2.03).

Exemple de script:

```
'VERSION(0)'
if result < 2.00 then do
'MESSAGE("Cette version du programme"+CHR(10)+"ne convient pas.")'
exit
end
...
```

### 1.317 Fonction ARexx VSCALE

VSCALE(numéro\_objet)

Cette fonction renvoie la valeur de l'échelle verticale de l'objet spécifié. La valeur de l'argument doit être comprise entre 1 et OBJECTS.

Si l'argument est nul c'est la valeur de placement courante qui est renvoyée (v2.07).

---

Voir aussi : HSCALE

### 1.318 Fonction ARexx WHEIGHT

WHEIGHT Version 2.05 (argument supprimé)

Renvoie la hauteur utile de la fenêtre spécifiée.  
La valeur renvoyée correspond en fait à la largeur du document, en pixels.

Voir aussi : WWIDTH, MESURE, WINDOW, SCREEN, DIMSHEET

### 1.319 Fonction ARexx WHILE

WHILE(fin,action1,...)

Cette fonction est identique à la fonction FOR, il lui manque juste le premier argument. L'initialisation des variables ou des cellules testées aura donc du être effectuée auparavant.

Voir aussi : SECURITY

### 1.320 Fonction ARexx WIDTH

WIDTH(numéro\_objet)

Renvoie la largeur de l'objet spécifié, en pixels.

Le numéro d'objet doit avoir sa valeur comprise entre 1 et OBJECTS, s'il est nul c'est la largeur du bloc sélectionné qui est renvoyée (ou -1 s'il n'y a aucun élément sélectionné).

Voir aussi : HEIGHT

### 1.321 Fonction ARexx WINDOW

WINDOW(x, y, largeur, hauteur)

Cette fonction permet de redimensionner la fenêtre courante, ou de la déplacer.

Les deux premiers arguments correspondent aux coordonnées du coin supérieur gauche de la fenêtre dans l'écran. Les deux arguments suivants déterminent ses dimensions.

Attention: cette fonction ne fonctionne que pour les fenêtres

---

"normales". Pour les fenêtres réduites, seules les coordonnées x et y peuvent être modifiées (pour déplacer les fenêtres). Les fenêtres cachées ne peuvent naturellement pas être modifiées, mais il suffit de spécifier la commande WINDOW(-1,-1,-1,-1) pour permettre leur réouverture. Si vous ne voulez pas modifier un ou plusieurs de ces paramètres et que vous ne connaissiez pas leur valeur, donnez-leur une valeur négative (-1). Si les arguments ont des valeurs trop grandes ou aberrantes, le programme essaiera d'y remédier au mieux. La valeur renvoyée est égale au nombre de valeurs ayant été prises en compte.

Exemples :

WINDOW(0,0,-1,-1) déplace la fenêtre en haut à gauche de l'écran sans modifier ses dimensions.

WINDOW(-1,-1,200,50) change les dimensions de la fenêtre

Voir aussi : MESURE

### 1.322 Fonction ARexx WRITE

WRITE("chaîne", x, y)

Écriture d'une chaîne de caractères à l'emplacement spécifié par les arguments x et y.

Si les valeurs de x et de y sont nulles, la chaîne spécifiée est placée sous le curseur, celui-ci est placé en mode placement de texte. L'utilisateur peut alors cliquer où il veut pour le placer (version 1.3).

L'échelle verticale et l'échelle horizontale courantes sont prises en compte pour déterminer les dimensions de cet élément (voir SETSCALE, ROTATE, SYMMETRY).

Si la fonction réussit, elle renvoie le numéro de l'objet qui a été placé, sinon elle renvoie 0.

Voir aussi : READTEXT.

### 1.323 Fonction ARexx WTOBACK

WTOBACK Version 2.05 (argument supprimé)

Envoie la fenêtre courante en arrière-plan.

Voir aussi : WTOFRONT, STOBACK, STOFront

### 1.324 Fonction ARexx WTOFRONT

---

WTOFRONT      Version 2.05 (argument supprimé)

Envoie la fenêtre courante à l'avant-plan.

Voir aussi : WTOBACK

### 1.325 Fonction ARexx WWIDTH

WWIDTH      Version 2.05 (argument supprimé)

Renvoie la largeur utile de la fenêtre courante.  
La valeur renvoyée correspond en fait à la hauteur du document,  
en pixels.

Voir aussi : WHEIGHT, MESURE, WINDOW, SCREEN, DIMSHEET

### 1.326 Fonction ARexx ZOOM

ZOOM(rapport)

Cette commande permet de connaître et de modifier le rapport de zoom définissant l'affichage.

Pour le modifier, spécifiez une valeur de l'argument comprise entre 25 (vue d'ensemble, la taille est alors réduite à 25% de la taille normale) et 1000 (zoom avec grossissement égal à 10 fois).  
Si l'argument n'est pas compris entre ces deux valeurs la fonction n'a aucun effet.

La valeur renvoyée est la valeur du rapport en vigueur avant l'appel de la fonction.

### 1.327 Commandes clavier

Rappel: la plupart des raccourcis permettant l'accès aux menus peuvent être obtenus sans utiliser la touche Amiga de droite.

HELP      Lance le programme AmigaGuide, celui-ci charge le fichier d'aide AmiCAD.guide. Ce fichier guide doit se trouver dans le répertoire où est situé le programme AmiCAD, ou bien à l'endroit spécifié par le type d'outil HELPFIELD.

Commandes d'édition

DEL      Effacement du ou des objets sélectionnés.

FLÈCHES déplacement curseur

---



FLÈCHES DE DÉPLACEMENT (GAUCHE / DROITE / HAUT / BAS) :  
déplacement des éléments sélectionnés, pixel par pixel ou par  
n pixels (taille de la grille) si la touche SHIFT est aussi appuyée.

ALT FLÈCHE:

Édition des rayons de l'arc sélectionné, pixel par pixel ou par  
n pixels (taille de la grille) pixels si la touche SHIFT est aussi  
appuyée.  
Déplacement d'une extrémité de la ligne sélectionnée par pixel ou par  
pas de la grille si la touche SHIFT est sélectionnée.

CONTROL FLÈCHE:

Édition des angles de l'arc sélectionné, degré par degré ou par  
par variation de n pixels (taille de la grille) degrés si la touche  
SHIFT est aussi appuyée.  
Déplacement d'une extrémité de la ligne sélectionnée par pixel ou par  
pas de la grille si la touche SHIFT est sélectionnée.

CTRL : permet de désélectionner des éléments, si utilisée en même  
temps que le bouton gauche de la souris.

TAB : copie (clonage) des éléments sélectionnés.

ENTER: fixe l'élément en cours (analogue à un clic sur le bouton gauche  
de la souris) ou passe en mode édition si un élément est sélectionné.

À noter: vous pouvez programmer n'importe quelle combinaison de touches de  
fonction en utilisant la fonction MAP, le menu Préférences/Touches de fonction ←  
,  
ou bien en incluant une définition dans le fichier AmiCAD.keys à  
l'aide d'un simple éditeur de texte.

## 1.328 Utilisation des touches de fonction

Les dix touches de fonction peuvent être associées à des séquences de  
commandes comprenant les appels aux fonctions que vous aurez définis. Ces  
séquences programmées sont sauvegardées dans le fichier "s:AmiCAD.keys",  
à l'aide du menu Préférences/Touches/Sauver. Celles-ci seront alors  
rechargées lors de chaque lancement du programme.

La redéfinition de ces touches peut se faire en utilisant le menu  
Préférences/Touches de fonction.

## 1.329 gtlayout.library

L'auteur de la bibliothèque gtlayout.library est Olaf 'Olsen' Barthel.

Elle est disponible sur Aminet (dev/gui), sources comprises.

## 1.330 Aide en ligne

---

Une aide en ligne peut être obtenue à tout moment, à l'aide du fichier AmiCAD.guide. Ce fichier doit se situer dans le même répertoire que le programme ou bien à un emplacement et un nom spécifiés dans le type d'outil HELPFIELD de l'icône du programme (cette information est prise en compte lors du lancement du programme).

Pour lancer l'aide vous pouvez appuyer sur la touche HELP, à tout moment ou bien lors de la sélection d'un menu, ce qui vous permet d'avoir directement l'aide concernant ce menu. Vous pouvez aussi utiliser le menu Projet/Aide et donner le nom d'un noeud (node) du fichier d'aide. Vous pouvez ainsi trouver une aide très rapidement pour n'importe quelle fonction, en donnant son nom.

À noter que vous pouvez également provoquer l'apparition de l'aide sur une fonction donnée, si une erreur survient, provoquée par cette fonction. Appuyez simplement sur la touche HELP alors que la requête signalant l'erreur est encore affichée.

### 1.331 Quelques macros utiles

Les définitions suivantes peuvent être intégrées au fichier AmiCAD.keys, soit à l'aide d'un éditeur de texte, soit en les définissant sous AmiCAD (voir fonction MAP) puis en choisissant le menu Préférences/Touches de fonction, et Sauver.

Macro permettant de choisir un composant en tapant uniquement les premières lettres de son nom (voire seulement la première):  
IF((PP=ASKTEXT("Composant?", "")) <> "", PUTPART(PP, -1, -1):MENU("Placer"), 0)

Sélection de tous les éléments du document:  
MARKZONE(0, 0, WWIDTH-1, WHEIGHT-1)

Chargement d'une bibliothèque de composants à la demande, sans passer par la requête gtlayout:

```
LOADLIB(ASKTEXT("Bibliothèque à charger?", ""))  
ou mieux:  
IF((LIB=ASKTEXT("Bibliothèque à charger?", "")) <> "", LOADLIB(LIB), 0)
```

Ajout du signe "ohm" à la valeur d'une résistance (la VALEUR de la résistance en question doit être seule sélectionnée):  
SETTEXT(FIRSTSEL, READTEXT(FIRSTSEL)+CHR(139))

Chargement d'un clip, passage de ce clip sous le curseur:  
LOADCLIP(1, "Additionneur"):MENU("Coller")

Pour lier un texte à un composant (en tant que valeur ou référence):  
LINKVAL(FIRSTSEL, PICKOBJ("Cliquez sur la valeur de ce composant"))  
LINKREF(FIRSTSEL, PICKOBJ("Cliquez sur la référence de ce composant"))

---

### 1.332 BUG(s) ? (mais oui ! sûrement... (malheureusement !))

Ce programme m'a demandé beaucoup de travail aussi je vous demanderais de bien vouloir être indulgent quant aux erreurs toujours possibles survenant lors de son utilisation. Je vous saurais gré de bien vouloir m'avertir si vous constatez une ou des anomalies.

L'utilisation du programme sous un écran CyberGraphics peut donner des résultats quelquefois bizarres (en fait la fonction de dessin en mode COMPLEMENT ne fonctionne alors pas ou très mal, pour les remplissages de surfaces, alors que le même programme fonctionne parfaitement sous les modes AGA). Ceci est dû au driver de la CyberVision 64 3D sous CyberGraphics, qui est bugué et toujours pas corrigé (il ne le sera sûrement jamais). Ce problème n'existe pas sous Picasso.

Il peut arriver qu'il survienne une erreur système lors de l'exécution d'un script ARexx, si une commande provoque une erreur. Ce problème est normalement évité en suivant la structure donnée en exemple dans le script squelette.AmiCAD pour les scripts ARexx.

L'écran AmiCAD est un écran public, ainsi vous pouvez ouvrir une (ou même plusieurs) autre(s) application(s) sur son écran. Cependant, fermez toujours ces autres fenêtres AVANT de quitter AmiCAD, sinon il peut y avoir quelques problèmes...

### 1.333 Historique

Version 2.08 31 mars 2001

- Ajout de l'objet Label (attribuable à un fil).
- Ajout des fonctions GETLABEL, LINKLAB, SETLABEL, FINDLAB, GETNET.
- Modification des fonctions SETREF, SETVAL (valeurs renvoyées).
- Ajout des fonctions ARRAYDIM, DIMARRAY, DELARRAY, GETARRAY et SETARRAY ( ← Traitement tableaux de données).
- Réécriture des scripts permettant la gestion des netlists et de la simulation (utilisent les nouvelles fonctions xxxARRAYxxx), nouveaux scripts (ChoixSignaux, ControlNets, CreateNets, LinkNets, Nets2File, VisuNets...)
- Modification gestion déplacements souris (prise en compte après le temps d'un double clic, comme défini dans les préférences du système, pour éviter les saut de composants intempestifs).

Version 2.07a 1er janvier 2001

- Modification bug traitement opérateur modulo (%)
- Quelques fôtes d'orthographe corrigées dans ce guide (comment ça, il n'en reste plus ?)

Version 2.07 12 novembre 2000

- Ajout fonction LANGUAGE (pour localisation des scripts).
- Correction bug chargement clip (hit Enforcer si composant non trouvé).
- Modification fonctions HSCALE et VSCALE pour pouvoir lire les échelles de placement courantes.
- Modification gestion présence boîte (plus précis).

---

Quelques améliorations au guide anglais (historique mis à jour, quelques fonctions ARexx ajoutées ou mises à jour).  
Correction d'un bug: lors du lancement du programme, si un argument était passé avec un nom de fichier qui n'existait pas, le programme ne signalait rien.  
Ajout du catalogue suédois (Henrik Isaksson: henisak@algonet.se).  
Correction bug sauvegarde touches de fonction (\_ en trop).

Version 2.06 1 novembre 2000

Correction bug gestion touches. Changement de version de la gtlayout.library ← (45).  
Nouveau bouton permettant une recherche des fonctions disponibles dans les requêtes de macro-commandes.  
Nouveaux boutons "Application" permettant de gérer l'application de l'édition des objets (Application aux objets sélectionnés, à tous les objets, aux objets de même type...)  
Ajout du catalogue finlandais (Joni Halme@: halmej@saunalahti.fi).

Version 2.05 14 avril 2000

Correction bug tirage fil (un fil était quelque fois tiré deux fois lors de la même opération).  
Correction bug impression (pouvait provoquer un plantage dans quelques rares cas, si la dernière ligne de la fenêtre était remplie).  
Impression par bandes (demande moins de mémoire chip, permet d'imprimer de plus grands documents, même avec une résolution élevée).  
Multitâche (chaque fenêtre est gérée par un process indépendant),  
modification de l'interface ARexx (due à la modification précédente) : quelques fonctions ont été supprimées (NEW, SELFIE, LOADKEYS, LOADPREF, MACRO, SAVEKEYS, SAVEPREF, SELSHEET, NBSHEET),  
quelques autres ont été ajoutées (NEWSHEET, NBSHEETS, REXXPORT) ou modifiées (CLOSE, MODIF, LOCK, UNLOCK, SAVEALL, OBJECTS, FONTNAME, FONTSIZE, WWIDTH, WHEIGHT, GETZONE, REQSHEET, WTOFRONT, WTOBACK).  
Certaines options disponibles sous Shell ont été supprimées : MACRO, SHEET\_WIDTH, SHEET\_HEIGHT, de même certains types d'outil ont aussi disparu : WINDOW, MACRO, GRIDSIZE. Ces paramètres peuvent être, pour la plupart, réglés dans les préférences.

Version 2.04 (28 novembre 99)

Correction bug placement valeur/référence à gauche d'un composant.  
Modification routine de tracé des arcs (plus lente mais résultats plus propres).  
Modification traitement requête pour gestion largeur traits sélectionnés (fautait si un fil spécial était édité). Extension possible pour boîtes, arcs, triangles, ellipses.  
Quelques modifications de ce guide (styles, corrections...)  
Correction bug programme AmiCAD2META (provoquait le crash du Miga lors de son lancement dans la version précédente, merci à Arno Richter).  
Ajout composant "Bornier" dans bibliothèque Symboles\_électriques.  
Nouveaux catalogues allemands pour les bibliothèques de symboles AmiCAD, divers et logiques (Merci à Gu0cky). Nouveau catalogue anglais pour les symboles logiques (myself).  
Modification gestion requêtes fichiers (rafraîchissement possible des fenêtres).  
Modification de la programmation des touches de fonction : on peut maintenant programmer ces touches avec n'importe quelle autre touche, avec ou sans ALT, SHIFT et (ou) CTRL. Nouveau menu et

nouvelle requête permettant la programmation de ces touches.  
Suppression de la macro ARexx MACRO.  
Nouvelle fonction REQLIST.

Version 2.03 (26 septembre 99)

Nouvelles requêtes permettant la modification des objets.  
Correction bug fonction MAP, nouveau script MapKey.

Version 2.02 (Août/Septembre 99)

Remplacement de la bgui.library par la gtlayout.library (code plus court et bien plus rapide).

Version 2.01 (Mai 99)

Correction de quelques bugs d'affichage après un défilement en mode "Auto-scroll".  
Modification de la fonction COORDS pour connaître les caractéristiques des angles et des ellipses (rayons, angles).  
Ajout gestion barre d'outils (TOOLBAR).  
Modification de la fonction MENU afin de traiter les noms de menus spécifiés en anglais, quelle que soit la locale utilisée.  
Modification des fonctions DELETE et SAVEIFF.  
Ajout fonction LIBRARY.  
Ajout menu Étirer pour changer l'échelle d'un groupe d'objets.

Version 2.00 (Mars 99)

Ajout fonctions CURSMODE, SETORTHO, SNAPGRID, GETLINE, GETZONE.  
Modification de la gestion de l'échelle des objets (maintenant exprimée en pourcentage: 100 donne une échelle normale).  
Modification du format des fichiers (utiliser le script ARexx ImportFile\_1.0 pour charger les anciens fichiers).  
Ajout zoom variable de 25 à 1000% (fonction ARexx: ZOOM).  
Correction bug mémoire allouée non rendue lors du chargement d'un clip.  
Correction d'un ÉNORME bug des fonctions SELSHEET et SELFIE entraînant le blocage des fenêtres sélectionnées.  
Ajout gestion du bouton central de la souris (choix document actif).  
Modification fonction REQFILE.

Version 1.6 (non distribuée)

Ajout spécification du type de connecteur lors de l'édition.  
Ajout gestion touche ENTRÉE (édition élément sélectionné, placement objet).  
Modification fonctions COL, LINE, HEIGHT et WIDTH pour connaître l'emplacement et les dimensions des éléments sélectionnés.  
Correction bug symétrie connecteurs, affichage rotations clips.  
Modification gestion connecteurs (symbole alterné, traitement symétrie).  
Modification de la fonction UNLINK (séparation valeur ou référence seule rendue possible).  
Correction bug impression jonctions carrées (plantage assuré dans la version 1.5...)  
Correction bug traitement symétrie des boîtes.  
Correction bug copie sans élément sélectionné (le clipboard courant était vidé).  
Ajout fonctions GETPOINT, ASKTEXT et ASKNUM.  
Suppression des fonctions ASK et GETPART (remplacées par ASKTEXT et PUTPART).

Version 1.5 2 novembre 1998

Ajout menus définis par l'utilisateur.  
Adaptation de l'utilitaire AmiCAD2META au format des bibliothèques,

---

bug rapporté par Henk Jonas.  
Ajout d'une requête permettant d'interrompre l'impression avant qu'elle ne soit terminée.  
Accélération de la sauvegarde (remarque de Sébastien VEYRIN-FORRER).  
Sauvegarde de l'état du menu "Préférences/Nom complet" lors d'une sauvegarde des préférences (rapport de David Beryl).  
Correction bug fonction PENWIDTH.  
Ajout gestion des jonctions carrées (utilisation du menu "Dessin/Alternar ↔ symbole").  
Problème ouverture requête pour choix composant en mode écran PAL résolu... ↔ (?)

#### Version 1.4 13 août 1998

Ajout fonctions ARexx GETDEVS, GROUP, UNGROUP, FINDOBJ, FINDLINE, PINCOL, PINLINE, PINNUM et DEVPINS.  
Bug Lock fixé (rapporté par Grzegorz Kraszewski).

#### Version 1.3 1er juin 1998

Correction du bug de la fonction "Restaurer" (enfin... ?)  
Corrections bugs traitement boîtes (impression et clips)...  
Correction bug fonction SETPINS. Ajout fonction BOX.  
Modification des fonctions WRITE, INPUT et OUTPUT.

#### Version 1.2 12 avril 1998

Corrections bugs : fonction DATE, utilisation fonctions utilisateur quelquefois défaillantes (DEF), fonctions SETDEV, SETGRID, REMLIB...  
Ajout fonctions SAVEALL et GETCOLOR.  
Amélioration de la gestion du menu Symétrie. Ajout du traitement des opérations de rotation, symétrie, agrandissement et diminution sur le clip en cours.  
Passage de l'écran de 8 à 16 couleurs: les éléments sont maintenant dessinés de couleurs différentes, selon leur type.  
Ajout de l'objet rectangle (pas de pointillés pour l'instant).  
Ajout d'un menu dans les préférences pour garder ou non le nom complet du fichier dans la barre de titre. Modification des indications portées dans la barre de titre (suggestion de Sébastien VEYRIN-FORRER).  
Amélioration du script Grille (création d'une grille dans le rectangle défini par l'utilisateur).  
Ajout du catalogue italien (par Massimo Basso), nouvelle version du catalogue espagnol (Benjamin Morente).  
Localisation des bibliothèques de symboles.

#### Version 1.1 8 mars 1998

Remplacement du fichier de configuration Configuration.AmiCAD par le fichier AmiCAD.prefs (plus besoin d'utiliser la ConfigFile.library, cependant les nouveaux fichiers de configuration ne sont pas compatibles avec les anciens).  
Amélioration du script d'installation.  
Écriture des catalogues allemand (par Henk Joans), tchèque (Vit Sindlar) et espagnol (Benjamin Morente).  
Modification ouverture écran (écran de dimensions égales à celui du Workbench par défaut).  
Correction de quelques petits bugs (largeur des boutons dans les requêtes, largeur des fenêtres réduites, gestion macro ARexx SYMMETRY) ainsi que de quelques autres plus importants (macro LOADCLIP).  
Ajout tooltypes SHEET\_WIDTH et SHEET\_HEIGHT.  
Quelques optimisations du code (programme plus court).

---

Ajout édition largeur trait composants, textes, ellipses, arcs, etc.  
Modification gestion des flèches curseur pour déplacement des objets  
et édition arcs et ellipses. Modification de la gestion des groupes.  
Ajout des fonctions AREXX SETPINS et COORDS.  
Suppression de l'utilisation de la ConfigFile.library, remplacée par  
des routines internes, plus courtes.  
Correction bug couleurs écran AGA.  
Suppression des "requesters", remplacés par des fenêtres...

Version 1.00 18 janvier 1998

Première version, écrite pour système 3.0. 1ère diffusion sur Aminet.

### 1.334 Aidez-moi!

Ce programme peut naturellement être très amélioré, je manque  
pour cela de documentations et surtout de temps pour développer  
toutes mes idées. Aussi si vous pouvez répondre à certaines de mes  
attentes, merci de me contacter.

Pour le moment je recherche de la documentation sur les points  
suivants:

- comment utiliser des polices vectorielles (CompuGraphics ou  
Adobe...) comme ProPage ou WordWorth,
- etc... (toute suggestion de votre part)

Merci pour votre collaboration.

### 1.335 AMÉLIORATIONS POSSIBLES

Les améliorations suivantes seront faites si le besoin s'en fait  
sentir et si j'en ai le temps (et aussi si j'en ai le courage !):

- choix de l'icône créée par le programme lors d'une sauvegarde,
- amélioration des messages d'erreurs, souvent imprécis,
- amélioration de la fonction MAP (visualisation et choix des macros  
déjà définies),
- choix et visualisation des macros dans une liste,
- marquage d'emplacements dans le schéma pour les retrouver  
rapidement (pratique dans un schéma de grande taille),
- requête permettant le choix d'un objet quand il y a superposition,
- écriture d'un éditeur de bibliothèque,
- et plus encore... (voir HELP!)

Si vous souhaitez traduire cette documentation ou bien le  
fichier catalogue, merci de me les communiquer afin qu'ils soient  
distribués avec le programme.

### 1.336 Traduction

---

Le programme est écrit en anglais, j'ai moi-même réalisé le catalogue français.

Le catalogue allemand a été réalisé par Henk Jonas:  
E-mail: subvcbhd@calvados.zrz.TU-Berlin.DE

Le catalogue tchèque a été réalisé par Vit Sindlar:  
E-mail: SINDLAR@jackal.cis.vutbr.cz

Le catalogue espagnol a été réalisé par Benjamin Morente:  
E-mail: ackman@mx3.redestb.es

Le catalogue italien a été réalisé par Massimo Basso:  
E-mail: cralex@amiga.dei.unipd.it

Le catalogue slovène a été réalisé par Daniel Krstic:  
E-mail: danny.k@www.comtron.si

Le catalogue finlandais a été réalisé par Joni Halme:  
E-mail: halmej@saunalahti.fi

Le catalogue suédois a été réalisé par Henrik Isaksson:  
E-mail: henisak@algonet.se.

Le guide anglais a été (beaucoup) amélioré par Alan Clarke:  
E-mail: 4wd@connexus.net.au

Si vous voulez traduire le catalogue dans une autre langue, merci de bien vouloir me l'envoyer.

Je recherche des volontaires pour traduire la doc AmigaGuide en anglais ou en allemand... Tous les bénévoles sont les bienvenus. Attention: les noms des nodes ne doivent pas être modifiés pour que le programme puisse correctement utiliser l'aide sous AmigaGuide.

## 1.337 AmiCAD2META

AmiCAD2META est un petit programme permettant de transformer un fichier au format AmiCAD (c'est à dire sauvé à l'aide de AmiCAD) en un format spécial utilisé par le programme MetaView et la bibliothèque amigametaformat.library. Ces deux programmes ont été écrits par Henk Jonas et permettent de sauver les fichiers AMF aux formats vectoriels supportés par cette bibliothèque. Pour l'instant les formats suivants sont supportés:

- WMF
- DR2D
- CGM
- GEM
- EPS
- AI
- HPGL
- ILBM

Il vous faudra obtenir ces programmes sur Aminet pour pouvoir les utiliser (gfx/conv/MetaView, util/dtype/DT\_MetaView, util/libs/amf\_library),

---



ceux-ci n'étant pas distribués avec AmiCAD. Vous pouvez aussi écrire à Henk Jonas pour avoir des précisions:  
E-mail: subvcbhd@calvados.zrz.TU-Berlin.DE

Pour utiliser AmiCAD2Meta vous pouvez utiliser le script ARexx Conv2Meta depuis AmiCAD ou bien directement appeler AmiCAD2Meta dans une fenêtre Shell, la syntaxe d'appel de ce programme est la suivante:

```
AmiCAD2Meta FROM/A,TO/K,FORCE/S,LIBS/K,VERBOSE/S,QUIET/S,PENWIDTH/N
```

Le premier argument est obligatoire (FROM), il spécifie quel est le fichier à traiter. Il doit s'agir d'un fichier préalablement créé lors d'une sauvegarde par AmiCAD. Le mot clé FROM peut être omis.

Exemples: AmiCAD2Meta Work:AmiCAD/Schémas/Logo  
AmiCAD2Meta FROM Work:AmiCAD/Schémas/Logo

Le second argument (TO) spécifie le nom du fichier AMF destination (au format Meta). S'il n'est pas donné aucune conversion ne se fera. Cela peut cependant être utile pour vérifier un fichier. Si ce fichier destination existe déjà, utilisez l'option FORCE pour qu'il soit ré-écrit.  
Exemple: AmiCAD2Meta ... TO ... FORCE

L'argument LIBS permet de préciser où sont situées les bibliothèques de symboles utilisées par AmiCAD.  
Exemple: AmiCAD2Meta ... LIBS Work:AmiCAD/Bibliothèques

L'argument VERBOSE permet d'afficher un certain nombre d'informations à l'écran alors que l'argument QUIET permet de n'avoir aucun affichage (utile dans un script ARexx). Ces deux derniers arguments doivent être naturellement être utilisés l'un sans l'autre. Si aucun d'entre eux n'est spécifié, seules les phases de traitement en cours sont signalées, sans précisions.  
Exemple: AmiCAD2Meta ... VERBOSE

Le dernier argument (PENWIDTH) permet d'élargir (éventuellement) les traits du multiple spécifié. Si cet argument n'est pas donné les traits sont sauvés avec la largeur présente sur le schéma, sinon leur largeur est multipliée par la valeur de cet argument. Hélas, de nombreux programmes ne traitent pas la largeur des traits mais les tracent toujours de la même largeur (Wordworth sous les formats CGM ou GEM, ProPage avec le format AI).  
Exemple: AmiCAD2Meta ... PENWIDTH=2

Exemples d'appel :

```
AmiCAD2Meta Work:AmiCAD/Schémas/Essai TO Work:MetaView/AmiCAD/Essai.amf LIBS  
Work:AmiCAD/Bibliothèques  
AmiCAD2Meta Work:AmiCAD/Schémas/Essai VERBOSE
```

## 1.338 L'auteur

Pour me contacter:

FLORAC Roland  
6 Rue des Chardonnerets  
Chez Corbin  
17610 Chaniers  
France  
Tél: 05 46 93 95 71

E-mail: roland.florac@fnac.net

## 1.339 Index

### A

ABS  
Aide en ligne  
Améliorations futures  
AppIcon  
ARC  
Arcs de cercles  
ARRAYDIM  
ASC  
ASK

### B

BLINK  
Boîtes  
BOX  
Bugs (?)

### C

CALL  
Chaînes de caractères  
CHR  
Les clips  
CLIPS  
CLIPPATH  
CLIPUNIT  
CLOSE  
COL  
Commandes~clavier  
Composants  
Connecteurs  
CONVERT  
COORDS  
COPY  
CURSMODE

---

## D

DATE  
DEF  
DELARRAY  
DELETE  
DEVPINS  
DIMARRAY  
DIMSHEET  
DRAW  
DRAWMODE  
Distribution  
Définition d'une fonction

## E

EDIT  
Ellipses  
ENDCOL  
ENDLINE  
ELLIPSE  
EXEC

## F

FILENAME  
FILEPART  
FINDLAB  
FINDLINE  
FINDOBJ  
FINDPART  
FINDREF  
FINDVAL  
FIRSTSEL  
Fonctions utilisateur  
FONTNAME  
FONTSIZE  
FOR

## G

GETARRAY  
GETCOLOR  
GETDEVS  
GETLABEL  
GETLINE  
GETNET  
GETPART  
GETPOS  
GETREF  
GETVAL  
GETZONE

GRIDSIZE  
GROUP  
GTLAYOUT

H

HEIGHT  
HELP  
HELPPFILE  
HISTORIQUE  
HSCALE

I

IF  
INIT  
INPUT  
Installation du programme

J

JUNCTION

L

Labels  
L'auteur  
Lancement du programme  
LANGUAGE  
LEN  
Les menus  
Les nombres  
LIBRARY  
LIBS  
LIBSPATH  
LINE  
LINKLAB  
LINKREF  
LINKVAL  
Liste alphabétique des fonctions ARexx  
Liste thématique des fonctions ARexx  
LOAD  
LOADCLIP  
LOADLIB  
LOCK

M

MACRO  
Type d'outil MACRO  
Macro-commande  
MAP  
MARK

---

MARKZONE  
MENU  
Menu Dessin  
Menu Édition  
Menu Macros  
Menu Projet  
Menu Préférences  
MESSAGE  
MESURE  
MODIF  
MOVE

N

NBSHEET  
NBSHEETS  
NEW  
NEWSHEET  
NEXTSEL

O

OBJECTS  
OPEN  
OUTPUT

P

Palette  
PARTNAME  
PASTE  
PENWIDTH  
PICKOBJ  
PINCOL  
PINLINE  
PINNUM  
Port ARexx  
PRINT  
PUTPART

R

READCONV  
READDEF  
READDEV  
READMAP  
READTEXT  
Rectangles  
REMLIB  
REQFILE  
REQLIST  
REQSHEET  
REQUEST  
RESET

REXEXPORT  
ROTATE

## S

SAVE  
SAVEALL  
SAVEIFF  
SAVECLIP  
SAVECOPY  
SAVEICON  
SAVEKEYS  
SAVEPREF  
SCREEN  
Scripts ARexx  
SCRMODE  
SECURITY  
SELECT  
SELSHEET  
SETARRAY  
SETCOLOR  
SETDEV  
SETGRID  
SETLABEL  
SETORTHO  
SETPINS  
SETREF  
SETSCALE  
SETTEXT  
SETVAL  
SGN  
SHEET\_HEIGHT  
SHEET\_WIDTH  
SHEIGHT  
SNAPGRID  
STARTUP  
STOBACK  
STOFRONT  
STR  
SWIDTH  
SYMMETRY

## T

TEST  
Textes  
TIME  
TITLE  
TOOLBAR  
Touches  
Triangles  
TXHEIGHT  
TXWIDTH  
TYPE

## U

UNGROUP  
UNLINK  
UNLOCK  
UNMAP  
UNMARK

## V

VAL  
Valeurs/Types  
Valeurs numériques  
Variables  
VERSION  
VSCALE

## W

WHEIGHT  
WHILE  
WIDTH  
WINDOW  
WRITE  
WTOBACK  
WTOFRONT  
WWIDTH

## X

X\_ICON

## Y

Y\_ICON

## Z

ZOOM

---