

### PROGRAMMER'S REFERENCE CARD

OP	HEX	DECIMAL	ASCII	
AFILL1	68	104	h	Write all pixels around P1 that have same color, using current color.
AFILL2	69	105	i	<b>EDGE</b> Write all pixels around P1 with current color, until edge.
ARC	62	98	b	<b>LO LENGTH</b> <b>HI LENGTH</b> Draw arc: center P1, start P2, $0 < \text{length} < 2047$ .
BLANK	4B	75	K	<b>ON(1)/OFF(0)</b> Blank display for faster writing.
BLINK	4D	77	M	<b>ON(1)/OFF(0)</b> Toggle highest plane (7 in Ω440, 3 in Ω420).
CHAR	6B	107	k	<b>CHAR 1</b> ... <b>CHAR N</b> <b>ESC</b> Text draw, using SETCSZ and SETORN.
CLEAR	60	96	,	Write entire screen with current color.
CMAP	51	81	Q	<b>PIXEL VALUE</b> <b>RED</b> <b>GREEN</b> <b>BLUE</b> Pixel value = color map address. (Note 1)
COMPDR	72	114	r	Draw vector from P1 to P2, complement pixels; WRMASK and SETCOL not used.
CRTWR	46	70	F	<b>REGISTER</b> <b>DATA</b> Write CRT controller register.
CSPACE	48	72	H	<b>LO ΔX</b> <b>HI ΔX</b> <b>LO ΔY</b> <b>HI ΔY</b> Set auto increment between characters.
CURS	71	113	q	Draw crosshair cursor at P1, using complement draw.
DRAW	61	97	a	Draw vector from P1 to P2, using WRMASK, SETCOL and PATTERN.
FFILL	65	101	e	Fill rectangle defined by P1, P2 diagonal. Fast, but no patterns allowed.
FSIZE	49	73	l	<b>ΔX</b> <b>ΔY</b> Set font size. $0 < \Delta X \leq 8$ , $0 < \Delta Y \leq 16$
GRAFIN	4A	74	J	<b>MODE</b> (Note 2)
INIT	5E	94	Λ	Initialize control functions to default values.
MOVP1	52	82	R	<b>LO X</b> <b>HI X</b> <b>LO Y</b> <b>HI Y</b> $0 \leq X, Y \leq 1024$
MOVP2	53	83	S	<b>LO X</b> <b>HI X</b> <b>LO Y</b> <b>HI Y</b> $0 \leq X, Y \leq 1024$
PATTERN	50	80	P	<b>TYPE</b> (Note 3)
PIXBLT	70	112	p	<b>LO ΔX</b> <b>HI ΔX</b> <b>LO ΔY</b> <b>HI ΔY</b> <b>DIRECTION</b> (Note 4)
POLYC	44	68	D	Close polygon substructure.
POLYF	67	103	g	Fill polygon defined by POLYS, POLYV, ..., POLYV.
POLYM	45	69	E	<b>LO X</b> <b>HI X</b> <b>LO Y</b> <b>HI Y</b> Add vertex to polygon; invisible edge.
POLYO	66	102	f	Draw edge to polygon defined by POLYS, POLYV, ..., POLYV.
POLYS	56	86	V	Initialize polygon data structure.
POLYV	57	87	W	<b>LO X</b> <b>HI X</b> <b>LO Y</b> <b>HI Y</b> Add vertex to polygon; visible edge.
PPAN	5B	91	[	Set origin of display (upper left corner) to P1.
RDCONF	5D	93	]	Controller returns two bytes. (Note 5)
RDMASK	4C	76	L	<b>MASK</b> 0 = all planes off; 255 = all planes on.
RDR	6E	110	n	Read rectangle defined by P1, P2 diagonal. (Note 6)
RECT1	63	99	c	Outline rectangle defined by P1, P2 diagonal.
RECT2	64	100	d	Fill rectangle defined by P1, P2 diagonal.
RLFILL	6A	106	j	<b>LO ΔX</b> <b>HI ΔX</b> Write ΔX successive pixels from P1. $0 \leq \Delta X \leq 1023$
RMOVP1	54	84	T	<b>LO ΔX</b> <b>HI ΔX</b> <b>LO ΔY</b> <b>HI ΔY</b> Relative move P1. $-2048 \leq \Delta X, \Delta Y \leq 2047$
RMOVP2	55	85	U	<b>LO ΔX</b> <b>HI ΔX</b> <b>LO ΔY</b> <b>HI ΔY</b> Relative move P2. $-2048 \leq \Delta X, \Delta Y \leq 2047$
RPIXEL	6C	108	l	Read pixel at P1.
SETCOL	4E	78	N	<b>COLOR</b> Set current color (0 to 255 in Ω440, 0 to 15 in Ω420).
SETORN	59	89	Y	<b>DIRECTION</b> Set character orientation. (Note 7)
SETCSZ	58	88	X	<b>X SCALE</b> <b>Y SCALE</b> Set character size. $0 \leq \text{scale} \leq 255$ for times 1, 2, ..., 256.
SIGRD	5C	92	\	Read polynomial signature from display (two bytes).
SYNCH	5F	95	-	<b>WAIT COUNT</b> Wait until the specified number of fields has occurred.
SZCUR	47	71	G	<b>LO ΔX</b> <b>HI ΔX</b> <b>LO ΔY</b> <b>HI ΔY</b> Cursor dimensions = $2\Delta X + 1$ wide by $2\Delta Y + 1$ high. $\Delta X, \Delta Y \leq 4095$
WPIXEL	6D	109	m	Write pixel at P1, using current color.
WRMASK	4F	79	O	<b>MASK</b> Set write mask. INIT sets to 255 (all planes written).
WRR	6F	111	o	<b>PIXEL 1</b> <b>PIXEL 2</b> ... <b>PIXEL N</b> Write rectangle defined by P1, P2 diagonal. (Note 6)
XDRAW	73	115	s	Draw vector from P1 to P2; write = present value EXOR current color.
ZOOM	5A	90	Z	<b>SCALE</b> Magnify memory relative to display origin. $0 \leq \text{scale} \leq 15$ for times 1, 2, ..., 16.

## OP CODES IN HEX ORDER

HI HEX

	4	5	6	7
0		PATERN	CLEAR	PIXBLT
1		CMAP	DRAW	CURSOR
2		MOVP1	ARC	COMPDR
3		MOVP2	RECT1	XDRAW
4	POLYC	RMOV1	RECT2	
5	POLYM	RMOV2	FFILL	
6	CRTWR	POLYS	POLYO	
7	SZCUR	POLYV	POLYF	
8	CSPACE	SETCSZ	AFILL1	
9	FSIZE	SETORN	AFILL2	
A	GRAFIN	ZOOM	RFILL	
B	BLANK	PPAN	CHAR	
C	RDMASK	SIGRD	RPIXEL	
D	BLINK	RDCONF	WPIXEL	
E	SETCOL	INIT	RDR	
F	WRMASK	SYNCH	WBR	

OF  
TILE

## NOTE 3: PATTERNS

**LINES (bit 3=1)**

**AREA (bit 3=0)**



### PATTERN ARGUMENT

	MODE		NORM	LINE	PATTERN
	X	Y	INV	AREA	
X	0	X			FOREGROUND: Write All Planes
X	1	X			FOREGROUND: Use Write Mask
1	X	X			BACKGROUND: Zero, Use Mask
0	X	0			BACKGROUND: Zero All Planes
0	X	1			BACKGROUND: No Change

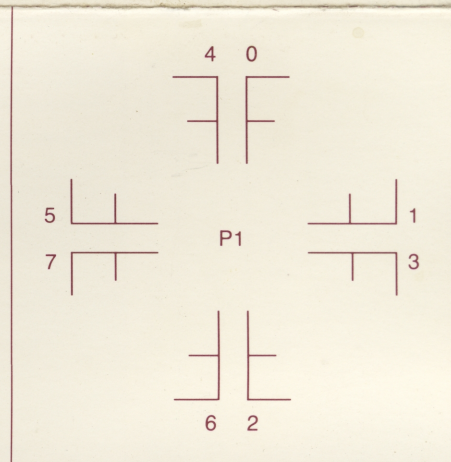
## NOTE 5: CONFIGURATION BYTE 1

Bit	IF = 0	IF = 1
0	33 Hz	60 Hz
1	0: 4 Planes	
2	1: 8 Planes	
3	2-7 Reserved	
4	No GRAFIN	GRAFIN

## NOTE 6: RECTANGLE SIZE, IN PIXELS

$$ABS(P1X - P2X) * ABS(P1Y - P2Y)$$

## NOTE 7: CHARACTER ORIENTATION



## NOTE 1: DEFAULT COLOR MAP

Pixel Value	R/R Max.	G/G Max.	B/B Max.
0	0	0	0
1	0	0	0
2	1.0	0	0
3	1.0	0	0
4	0	1.0	0
5	0	1.0	0
6	1.0	1.0	0
7	1.0	1.0	0
8	0	0	1.0
9	0	0	1.0
A	1.0	0	1.0
B	1.0	0	1.0
C	0	1.0	1.0
D	0	1.0	1.0
E	1.0	1.0	1.0
F	1.0	1.0	1.0

Above for 4 plane Ω420; for Ω440, use the following:  
 $P = 224\hat{R} + 28\hat{G} + 3\hat{B}$  where P is pixel value (base 10),  
 $\hat{R}$  and  $\hat{G}$  are fractions of maximum chosen from (0, 1/8, ... 7/8) and  $\hat{B}$  from (0, 1/4, 2/4, 3/4)

## NOTE 4: PIXBLT DIRECTION BYTE

Bit	Meaning
0	SWAP X,Y AXES
1	DESTINATION, Y DEC
2	DESTINATION, X DEC
3	SOURCE, Y DEC
4	SOURCE, X DEC

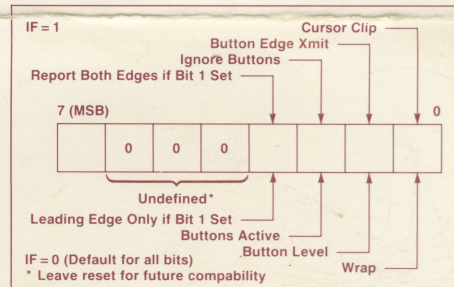
## NOTE 2: GRAFIN ARGUMENTS

0	Initialize Scale Factors and Delimiter			
1	Set Cursor Tracking Mode			
2	Set Transparent Mode			
3	LO X OFFSET	HI X MULTIPLIER	LO Y OFFSET	HI Y MULTIPLIER
4	DELIMITER Set Transparent Mode Delimiter			
5	Sample Screen Position			
6	Sample Tablet Position			
7	MODE ARG Write Mode Register (See GRAFIN MODE REGISTER chart)			

If Bit 7 = 0, bits 2, 1, and 0 point to the target bit in the Mode Register, which is to be set to the value of Bit 4.

If Bit 7 = 1, then bits 6, 5, ..., 0 overwrite the corresponding positions in the Mode Register.

## GRAFIN MODE REGISTER



### PROGRAMMER'S REFERENCE CARD

OP	HEX	DECIMAL	ASCII		
AFILL1	68	104	h	Write all pixels around P1 that have same color, using current color.	
AFILL2	69	105	i	EDGE	Write all pixels around P1 with current color, until edge.
ARC	62	98	b	LO LENGTH	HI LENGTH Draw arc: center P1, start P2, $0 < \text{length} < 2047$ .
BLANK	4B	75	K	ON(1)/OFF(0)	Blank display for faster writing.
BLINK	4D	77	M	ON(1)/OFF(0)	Toggle highest plane (7 in Ω440, 3 in Ω420).
CHAR	6B	107	k	CHAR 1 ... CHAR N	ESC Text draw, using SETCSZ and SETORN.
CLEAR	60	96	,	Write entire screen with current color.	
CMAP	51	81	Q	PIXEL VALUE	RED GREEN BLUE Pixel value = color map address. (Note 1)
COMPDR	72	114	r	Draw vector from P1 to P2, complement pixels: WRMASK and SETCOL not used.	
CRTWR	46	70	F	REGISTER	DATA Write CRT controller register.
CSPACE	48	72	H	LO ΔX	HI ΔX LO ΔY HI ΔY Set auto increment between characters.
CURS	71	113	q	Draw crosshair cursor at P1, using complement draw.	
DRAW	61	97	a	Draw vector from P1 to P2, using WRMASK, SETCOL and PATTERN.	
FFILL	65	101	e	Fill rectangle defined by P1, P2 diagonal. Fast, but no patterns allowed.	
FSIZE	49	73	l	ΔX	ΔY Set font size. $0 < \Delta X \leq 8$ , $0 < \Delta Y \leq 16$
GRAFIN	4A	74	J	MODE	(Note 2)
INIT	5E	94	^	Initialize control functions to default values.	
MOVP1	52	82	R	LO X	HI X LO Y HI Y $0 \leq X, Y \leq 1024$
MOVP2	53	83	S	LO X	HI X LO Y HI Y $0 \leq X, Y \leq 1024$
PATTERN	50	80	P	TYPE	(Note 3)
PIXBLT	70	112	p	LO ΔX	HI ΔX LO ΔY HI ΔY DIRECTION (Note 4)
POLYC	44	68	D	Close polygon substructure.	
POLYF	67	103	g	Fill polygon defined by POLYS, POLYV, ..., POLYV.	
POLYM	45	69	E	LO X	HI X LO Y HI Y Add vertex to polygon; invisible edge.
POLYO	66	102	f	Draw edge to polygon defined by POLYS, POLYV, ..., POLYV.	
POLYS	56	86	V	Initialize polygon data structure.	
POLYV	57	87	W	LO X	HI X LO Y HI Y Add vertex to polygon; visible edge.
PPAN	5B	91	[	Set origin of display (upper left corner) to P1.	
RDCONF	5D	93	]	Controller returns two bytes. (Note 5)	
RDMASK	4C	76	L	MASK	0 = all planes off; 255 = all planes on.
RDR	6E	110	n	Read rectangle defined by P1, P2 diagonal. (Note 6)	
RECT1	63	99	c	Outline rectangle defined by P1, P2 diagonal.	
RECT2	64	100	d	Fill rectangle defined by P1, P2 diagonal.	
RLFILL	6A	106	j	LO ΔX	HI ΔX Write ΔX successive pixels from P1. $0 \leq \Delta X \leq 1023$
RMOVP1	54	84	T	LO ΔX	HI ΔX LO ΔY HI ΔY Relative move P1. $-2048 \leq \Delta X, \Delta Y \leq 2047$
RMOVP2	55	85	U	LO ΔX	HI ΔX LO ΔY HI ΔY Relative move P2. $-2048 \leq \Delta X, \Delta Y \leq 2047$
RPIXEL	6C	108	l	Read pixel at P1.	
SETCOL	4E	78	N	COLOR	Set current color (0 to 255 in Ω440, 0 to 15 in Ω420).
SETORN	59	89	Y	DIRECTION	Set character orientation. (Note 7)
SETCSZ	58	88	X	X SCALE	Y SCALE Set character size. $0 \leq \text{scale} \leq 255$ for times 1, 2, ..., 256.
SIGRD	5C	92	\	Read polynomial signature from display (two bytes).	
SYNCH	5F	95	-	WAIT COUNT	Wait until the specified number of fields has occurred.
SZCUR	47	71	G	LO ΔX	HI ΔX LO ΔY HI ΔY Cursor dimensions = $2\Delta X + 1$ wide by $2\Delta Y + 1$ high. $\Delta X, \Delta Y \leq 4095$
WPIXEL	6D	109	m	Write pixel at P1, using current color.	
WRMASK	4F	79	O	MASK	Set write mask. INIT sets to 255 (all planes written).
WRR	6F	111	o	PIXEL 1	PIXEL 2 ... PIXEL N Write rectangle defined by P1, P2 diagonal. (Note 6)
XDRAW	73	115	s	Draw vector from P1 to P2; write = present value EXOR current color.	
ZOOM	5A	90	Z	SCALE	Magnify memory relative to display origin. $0 \leq \text{scale} \leq 15$ for times 1, 2, ..., 16.

OP CODES IN HEX ORDER

HI HEX

	4	5	6	7
0		PATERN	CLEAR	PIXBLT
1		CMAP	DRAW	CURSOR
2		MOVVP1	ARC	COMPDR
3		MOVVP2	RECT1	XDRAW
4	POLYC	RMOVVP1	RECT2	
5	POLYM	RMOVVP2	FFILL	
6	CRTWR	POLYS	POLYO	
7	SZCUR	POLYV	POLYF	
8	CSPACE	SETCSZ	AFILL1	
9	FSIZE	SETORN	AFILL2	
A	GRAFIN	ZOOM	RLFILL	
B	BLANK	PPAN	CHAR	
C	RDMASK	SIGRD	RPIXEL	
D	BLINK	RDCONF	WPIXEL	
E	SETCOL	INIT	RDR	
F	WRMASK	SYNCH	WRP	

XMIT OR

NOTE 3: PATTERNS

**LINES (bit 3 = 1)**

**AREA (bit 3 = 0)**

NOTE 5: CONFIGURATION BYTE 1

Bit	IF = 0	IF = 1
0	33 Hz	60 Hz
1	0: 4 Planes	
2	1: 8 Planes	
3	2-7 Reserved	
4	No GRAFIN	GRAFIN

NOTE 6: RECTANGLE SIZE, IN PIXELS

$$ABS(P1X - P2X) * ABS(P1Y - P2Y)$$

NOTE 7: CHARACTER ORIENTATION

NOTE 1: DEFAULT COLOR MAP

Pixel Value	R/R Max.	G/G Max.	B/B Max.
0	0	0	0
1	0	0	0
2	1.0	0	0
3	1.0	0	0
4	0	1.0	0
5	0	1.0	0
6	1.0	1.0	0
7	1.0	1.0	0
8	0	0	1.0
9	0	0	1.0
A	1.0	0	1.0
B	1.0	0	1.0
C	0	1.0	1.0
D	0	1.0	1.0
E	1.0	1.0	1.0
F	1.0	1.0	1.0



PATTERN ARGUMENT

MODE	NORM INV	LINE AREA	PATTERN
X 0 X			FOREGROUND: Write All Planes
X 1 X			FOREGROUND: Use Write Mask
1 X X			BACKGROUND: Zero, Use Mask
0 X 0			BACKGROUND: Zero All Planes
0 X 1			BACKGROUND: No Change

NOTE 4: PIXBLT DIRECTION BYTE

Bit	Meaning
0	SWAP X, Y AXES
1	DESTINATION, Y DEC
2	DESTINATION, X DEC
3	SOURCE, Y DEC
4	SOURCE, X DEC

NOTE 2: GRAFIN ARGUMENTS

0	Initialize Scale Factors and Delimiter																								
1	Set Cursor Tracking Mode																								
2	Set Transparent Mode																								
3	<table border="1"> <thead> <tr> <th>LO</th> <th>HI</th> <th>FRAC</th> <th>INT</th> <th>LO</th> <th>HI</th> <th>FRAC</th> <th>INT</th> </tr> </thead> <tbody> <tr> <td colspan="4">X OFFSET</td> <td colspan="4">X MULTIPLIER</td> </tr> <tr> <td colspan="4">Y OFFSET</td> <td colspan="4">Y MULTIPLIER</td> </tr> </tbody> </table> Set Scale and Offset	LO	HI	FRAC	INT	LO	HI	FRAC	INT	X OFFSET				X MULTIPLIER				Y OFFSET				Y MULTIPLIER			
LO	HI	FRAC	INT	LO	HI	FRAC	INT																		
X OFFSET				X MULTIPLIER																					
Y OFFSET				Y MULTIPLIER																					
4	DELIMITER Set Transparent Mode Delimiter																								
5	Sample Screen Position																								
6	Sample Tablet Position																								
7	MODE ARG Write Mode Register (See GRAFIN MODE REGISTER chart)																								

If Bit 7 = 0, bits 2, 1, and 0 point to the target bit in the Mode Register, which is to be set to the value of Bit 4.

If Bit 7 = 1, then bits 6, 5, ..., 0 overwrite the corresponding positions in the Mode Register.

GRAFIN MODE REGISTER