

JBit IO (R)**Registers**

REQPUT	1	CONVIDEO	40(40)
REQEND	2	CONROW0	40(10)
REQRES	3	CONROW1	50(10)
REQPTRLO	4	CONROW2	60(10)
REQPTRHI	5	CONROW3	70(10)
ENABLE	16	LID	80
FRMFPS	17	LCTL	81
FRMDRAW	18	LX	82
GKEY0	19	LY	83
GKEY1	20	SFRAME	84
RANDOM	23	STRANSFM	85
KEYBUF	24(8)	SCWITH	86
CONCOLS	32	TCOLLO	87
CONROWS	33	TCOLHI	88
CONCX	34	TROWLO	89
CONCY	35	TROWHI	90
CONCCHR	36	TCELL	91
CONCFG	37	REQDAT	96(32)
CONCBG	38		

Constants**ENABLE (Bitmask)**

BGCOL	1	BGIMG	2
CONSOLE	4	LAYERS	8

Standard Palette

BLACK	0	ORANGE	8
WHITE	1	BROWN	9
RED	2	LIGHTRED	10
CYAN	3	GRAY1	11
PURPLE	4	GRAY2	12
GREEN	5	LIGHTGREEN	13
BLUE	6	LIGHTBLUE	14
YELLOW	7	GRAY3	15

Line Art (Bitmask)

TOP	1	BOTTOM	8
LEFT	2	RIGHT	4
CDOT	128		

Datatypes

U8	1	I8	2
U16	3	I16	4

LCTL - Layer Control (Bitmask)

SHIFTX0	1	SHIFTX1	2
SHIFTY0	4	SHIFTY1	8
PXLCOLL	16	ENABLE	128

STRANSFM - Sprite Transformation

NONE	0	ROT90	5
ROT180	3	ROT270	6
MIRROR	2	MRROT90	7
MRROT180	1	MRROT270	4

GKEY0 - Game Keys 0 (Bitmask)

UP	2	LEFT	4
RIGHT	32	DOWN	64

GKEY1 - Game Keys 1 (Bitmask)

A	2	B	4
C	8	D	16
FIRE	1		

Requests

On the other side of this sheet you can find the syntax of the requests (and the corresponding results, when applicable). See the *bgcol1* and *bgcol2* demos for examples of how to send a request to the IO chip. After the request has been sent, REQRES contains 0 on success and 255 on failure (usually not tested). Results are available starting from REQDAT. Streaming requests (identified by a >) are not bounded. The other requests are bounded (255 bytes).

Optional values are delimited by [and]. (and) are used for grouping. * means repeat 0 or more times. + means repeat at least once. # means repeat with constraints. | means choice (priority is low). Datatype is U8 unless stated otherwise (by a tag preceded by :). For datatypes larger than 8 bits, the least significant byte comes first. Enumerated values are identified by C (for choice). Bitmasks are identified by O (for OR). Strings can be delimited by 0 (S0) or by the end of the request (S). When an argument has datatype T, the actual datatype is chosen by the user with DType (see constants above).

For the semantic of the requests, take a look at the demos or simply experiment using names as hints. Notes for the IPNGGEN request: using INDEXED_COLOR causes a PLTE chunk to be generated (a palette must be provided) and setting IDX0TRANSP causes a tRNS chunk to be generated. For more information see the PNG specification.

System

NOREQ(0)>: Dummy*
TIME(2): [RefTime=ABS [Fract=1000]]
 RefTime(C): ABS(1), RESET(2)
 Fract(C): 1(1), 10(2), 10(3), 1000(4)
 Result: Time:U64
LOADROM(6): Addr:U16 ResName:S0 [Offset:U16 Size:U16]
RSFORMAT(8): 121 33
RLOAD(9): Addr:U16 Size:U16 RecName:S0
RSAVE(10): Addr:U16 Size:U16 RecName:S0
RDELETE(11): RecName:S0
Display and Imaging
DPYINFO(16): -
 Result: Width:U16 Height:U16 ColorDepth
 AlphaDepth Flags
 Flags(O): ISCOLOR(128), ISMIDP2(64)
SETBGCOL(17): PaletteEntry | Red Green Blue
SETPAL(18)>: (Red Green Blue)*
SETBGIMG(19): ImageId
IDESTROY(20): ImageId
IDIM(21): MaxImageId
IINFO(22): ImageId
 Result: Width:U16 Height:U16 Flags
 Flags(O): ISMUTABLE(128)
ILOAD(23): ImageId ResName:S0
IDUMMY(24): ImageId Type=SIMPLE Width:U16 Height:U16 Bg Fg [Title:S]
IDUMMY(24): ImageId Type=SPRITE Width Height Frames Bg Fg [Title:S]
IDUMMY(24): ImageId Type=TILES Width Height Cols Rows Bg Fg (N Bg Fg)*
 Type(C): SIMPLE(1), SPRITE(2), TILES(3)
IPNGGEN(25)>: ImageId Width:U16 Height:U16 Depth ColorType Flag [MaxPaletteEntry (PaletteEntry | Red Green Blue)#] Data#
 ColorType(C): GRAYSCALE(0), TRUECOLOR(2), INDEXED_COLOR(3), GRAYSCALE_ALPHA(4), TRUECOLOR_ALPHA(6)
 Flags(O): IDXOTRANS(1), PALREF(2), ZOOM0(4), ZOOM1(8), ZOOM2(16)
IRAWRGBA(28)>: ImageId Width:U16 Height:U16 Flags (Red Green Blue Alpha)#
 Flags(O): ALPHA(128)

Layers (Game API)

LMVIEW(32): TiledLayerId | DType [X:T Y:T] Width:T Height:T
LMPOS(33): DType OX:T OY:T
LDESTROY(34): LayerId
LDIM(35): MaxLayerId
LTILED(36): TiledLayerId ImageId TWidth THeight NAnimTiles DType Cols:T Rows:T
LSPRITE(37): SpriteId ImageId [Width Height]
LSETPOS(38): LayerId DType X:T Y:T
LGETPOS(39): LayerId
 Result: X:I32 Y:I32
LMOVE(40): LayerId DType DX:T DY:T
LSETPRI(41): LayerId DType Priority:T
LGETPRI(42): LayerId
 Result: Priority:I32
LTLANIM(43): TiledLayerId AnimTile StaticTile
LTLFILL(44): TiledLayerId Tile [DType Col:T Row:T NumCols:T NumRows:T]
LTLPUT(45)>: TiledLayerId Col:U16 Row:U16 NumCols:U16 Tile*
LTLSCROLL(46): TiledLayerId ScrollType=0 DType Col:T Row:T NumCols:T NumRows:T DX:T DY:T
LSPCOPY(47): SpriteId TemplateSpriteId
LSPAPOS(48): SpriteId DType AbsX:T AbsY:T
LSPREFPX(49): SpriteId DType RefPixelX:T RefPixelY:T
LSPCLRCT(50): SpriteId DType CollRctOX:T CollRctOY:T CollRctWidth:T CollRctHeight:T
GAMESET(60): [(ImageId | TileSet=SILK) [Cols=0 Rows=0 [LayerId=1 [TWidth=0 THeight=0]]]]
 TileSet(C): SILK(255), FONT(254), MICRO(253)
 Result: Cols:U16 Rows:U16

Effects

FXTONE(64): Duration Frequency Volume
FXVIBRA(65): Duration
 Result: Supported
FXFLASH(66): Duration
 Result: Supported