



Fujitsu GDC Studio Version 1.0.0.0







Background Information :

- The latest version (described here) is: Version 1.0.0.0
- Extensions to the current preview are in progress
- Changes and / or modifications may occur without prior notice



Fujitsu GDC Studio Layout



Graphics Competence Center







Flexible Project File Architecture

- Fujitsu GDC Studio's functionality is configured by loading a chip specific project file (.gdcproj)
- Project files are currently available for :
 - MB88F332 'Indigo',
 - MB86928 'Ruby'
- Additional project files for future Fujitsu chips are planned
- Project files can also be saved / reloaded when a specific chip status should be recorded

Available Features

- GUI based Register Debugger and Register Sequencer
- GUI based Image Manager and Font Manager
- GUI based memory inspection and editing functions:
 - Memory Editor and Hex Dump
 - Flash Editor, Flash Dump and Programmer





Modular Hardware Abstraction

- Depending on the chip design there are one or more of the following hardware access types possible :
 - SPI Serial Peripheral Interface (connected via USB)
 - PCIe Peripheral Component Interconnect Express
- Additional interface support is possible but currently not planned
- Automatic Installation Setup Wizard available
- User Manual with detailed Information available online
- Release notes available online





🔗 Fujitsu GDC Studio							<u>? ×</u>
Eile Project Settings Help							
							+ T []
Begister Debugger	Name	Tupe	Rit Offeet	Rit Width	Field Value	Begister Value Error	
Stepper Motor Control		BW/	7	1	0		
A/D Converter		BW	8	1	ő		
- 🌛 Reload Timer Unit		BW	9	1	ő		•
🥹 I2C Unit		BW	10	1	ő		
		BW/	11	1	ő		
UART Unit		1199		1	, in the second s	0-0000000	
🥹 Pulse Generator		в	0	5	0	0.0000000	
- 🌛 Chip Control		BW.	5	1	0		
- 🌛 Remote Handler	4 MD 1 0	DW PU/	с С	2	0		
🤪 Configuration Fifo		POVD	0	1			
		Du/	0	1	0		
🍪 Clock Synthesis		nw BW	3		0		
General Purpose IO	✓ 515_1_0	BW	10	2	0		
- Grite Engine Ctrl	PAUS	HW	12	1	0		
- Grite Engine SAT	VINTE NIT	HW	13	1	U		
Sprite Engine SS		HW	14	1	U		
Cianabas Unit	BUSY	HW	15	1	U		
Color Lookura Tobla		_		_			
Dithoring Unit	D_7_0	н	U	8			
Dichenny Onic	D_9_8	В	8	2			
Display Controller	ADCT0_ADCT1		_	_		0x0000102C	
	ST_7_0	RW	0	8	44		
- A SPI Flash	ST_9_8	RW	8	2	0		
	CT_5_0	RW	10	6	4		
Command Sequencer	G ADECH_ADSCH					0x0000000	
E-C Register Sequencer	ANE_4_0	RW	0	5	0		
panel_init_ET057003DM6_320x240	• ANS_4_0	RW	8	5	0		
	A/D alternation of the						
spe_test_color_bars_320x240	A/D channel setting register						
	Hexadecimal			Deci	imal	Discrete Values	
spe_test_special_move_320x240	Bit Range 12 8 7				7	0x0	
Image Manager	Bit Width 5					0x1 🗖	
Font Manager						0x2	
Memory Editor	Reset Value UxU						
Memory Dump	Access Type RW 0				31		
						Indian MR88F3	32 🕢 .
						inaigo j Mboor a	





Covers the most essential hardware IP's of an available design

• e.g. MB88F332 'Indigo,, MB86928 'Ruby'

Detailed hardware information is visible for the selected ...

- IP / Component
- Address Block
- Register
- Register Field

Practical register/field reads and writes

- Depending on the register properties, various 'simple to use' controls appear
- The register access type is handled accordingly (R, RW, W, RSVD, ...)

Pop-up descriptions of the selected register fields are displayed





Hardware access status supported

- An 'Error' column reporting the status of each access to the corresponding field is displayed on the far right
- An additional 'Action Reporting Window' can be opened if detailled access information is needed

Hardware connect / disconnect button available

- Enables a user to use Fujitsu GDC Studio with a different parallel target application
- Prevents hardware access problems if GDC Studio and a separate application use the same driver interface
- Fujitsu GDC Studio may be disconnected or reconnected at any time





🔗 Fujitsu GDC Studio ? × File Project Settings Help 过 🛃 🗙 ±∓□ E- C Register Debugger Component Name Register Name Address Size Register Value Error 🔺 Stepper Motor Control 🗲 GeneralCtrl **Clock Synthesis** 0x00014000 32 bit 0x00000002 🌛 A/D Converter NominalFrequency Clock Synthesis 0x00014004 32 bit 0x00130200 • 🍓 Reload Timer Unit 🖕 НТР **Display Controller** 0x00034004 32 bit 0x01980000 🍓 I2C Unit 🔶 HDP **Display Controller** 0x00034008 32 bit 0x00000140 🍓 Sound Generator HSP_HSW_VSW **Display Controller** 0x0003400C 32 bit 0x03280161 🍓 UART Unit VTB. 0x00034010 0x01040000 **Display Controller** 32 bit 🍓 Pulse Generator VDP_VSP **Display Controller** 0x00034014 32 bit 0x00F000F4 \mathbf{O} 🍓 Chip Control 🔶 DСМО 0x80060004 **Display Controller** 0x00034000 32 bit 🌛 Remote Handler DLS **Display Controller** 0x00034210 32 bit 0x00000001 -Configuration Fifo 🔶 LAETC **Display Controller** 0x00034240 32 bit 0x80000000 Clock Modulator 6 🔶 PFD --**Display Controller** 0x00034364 32 bit 0x00000000 🍓 Clock Synthesis 🔶 Interrupt_Enable **Display Controller** 0x00034204 32 bit 0x00000100 🌛 General Purpose IO SPEDPAR Sprite Engine Ctrl 0x00020024 32 bit 0x00000140 🌛 Sprite Engine Ctrl ቀ DIR RBM CTRL 0x00000009 🍓 Sprite Engine SAT **Timing Controller** 0x0002852C 32 bit 🍓 Sprite Engine SS DIR_PIN0_CTRL **Timing Controller** 0x00028534 32 bit 0x00000090 Timing Controller 🔶 DIR_PIN1_CTRL **Timing Controller** 0x00028538 32 bit 0x00000090 ➡ 0 🍛 Signature Unit DIR_PIN2_CTRL 0x00000090 **Timing Controller** 0x0002853C 32 bit 🍛 Color Lookup Table DIR_PIN3_CTRL **Timing Controller** 0x00028540 32 bit 0x00000090 ۰ 🌛 Dithering Unit DIR_PIN4_CTRL Timing Controller 0x00028544 32 bit 0x00000090 Display Controller DIR_PIN5_CTRL **Timing Controller** 0x00028548 32 bit 0x00000090 🌛 DMA Controller 🛑 DIR PING CTRL 32 bit 0x00000090 **Timing Controller** 0x0002854C ÷ Memory Interface DIR_PIN7_CTRL **Timing Controller** 0x00028550 32 bit 0x00000090 🍓 SPI Flash DIR_PIN8_CTRL **Timing Controller** 0x00028554 32 bit 0x00000090 실 Run Length Decoder 🛑 DIR PIN9 CTRL **Timing Controller** 0x00028558 32 bit 0x00000091 Command Sequencer DIR_PIN10_CTRL **Timing Controller** 0x0002855C 32 bit 0x00000090 🖻 🦢 Register Sequencer ቀ DIR PIN11 CTRL **Timing Controller** 0x00028560 32 bit 0x00000090 panel init ET057003DM6 320x240 DIR PIN12 CTRL **Timing Controller** 0x00028564 32 bit 0x00100091 • spe_test_color_bars_320x240 1912 spe_test_single_color spe_test_special_blink_32x240 Sequence Execution Mode Delay spe_test_special_move_320x240 MB88F332 Chip Id Default, Independent Ŧ 🚞 Image Manager Default, Independent Codename Indigo 🦲 Font Manager Indigo, Direct Access 🛅 Memory Editor Version. 0.1 Indigo, Command List 🚞 Memory Dump Save Sequence Close Sequence Indigo MB88F332



Register Sequencer - Debug



🔗 Fujitsu GDC Studio ? × <u>File Project Settings</u> Help 过 🛃 🗙 ±∓□ E-/ Register Debugger Register Name Component Name Address Size Register Value Error 0 Stepper Motor Control 🔶 ChipInfo 0x00010000 32 bit 0x20080000 Chip Control 🌛 A/D Converter e. 🍓 Reload Timer Unit SPELUTS 0xFF000000 Sprite Engine Ctrl 0x00026000 32 bit 🍓 I2C Unit SPELUTS Sprite Engine Ctrl 0x00026004 32 bit 0x0000FF00 🍓 Sound Generator SPELUTS Sprite Engine Ctrl 0x00026008 32 bit 0x00FF0000 🍓 UART Unit **6**57 🍓 Pulse Generator InterruptStatusW0 32 bit 0x000001FF Signature Unit 0x0002A05C 0 🍓 Chip Control 🖕 ColourIndexElement Color Lookup Table 0x0002C000 32 bit 0x00000000 🌛 Remote Handler 💠 ColourIndexElement Color Lookup Table 0x0002C004 32 bit 0x00000000 Configuration Fifo 💠 ColourIndexElement Color Lookup Table 0x0002C008 32 bit 0x00000000 Clock Modulator 2 ٠ 🍓 Clock Synthesis Dither Control Dithering Unit 0x0002E000 32 bit 0x00000001 🍓 General Purpose IO \bigcirc Software Delay Element 100 🌛 Sprite Engine Ctrl H 🍓 Sprite Engine SAT 🍓 Sprite Engine SS 🔶 HDP -**Display Controller** 0x00034008 32 bit 0x0000027F Timing Controller HSP_HSW_VSW **Display Controller** 0x0003400C 32 bit 0x015F028F ⇒ 🍛 Signature Unit VTR 0x00034010 0x020C0000 **Display Controller** 32 bit 🌛 Color Lookup Table + VDP_VSP **Display Controller** 0x00034014 32 bit 0x01DF01E9 🍓 Dithering Unit Software Delay Element 200 🍓 Display Controller 🔶 Status **Display Controller** 0x00034200 32 bit 0x00000000 🌛 DMA Controller ÷ Memory Interface 🍓 SPI Flash 실 Run Length Decoder Command Sequencer X 🖻 🗁 Register Sequencer panel_init_ET057003DM6_320x240 spe_test_color_bars_320x240 spe_test_special_blink_32x240 Sequence Execution Mode Delay spe_test_special_move_320x240 Chip Id MB88F332 Default, Independent • →> Example Sequence Codename Indigo 🦲 Image Manager 🛅 Font Manager 0.1 Version 🚞 Memory Editor Save Sequence Close Sequence 🚞 Memory Dump Indigo MB88F332 \otimes





Easy creation of register sequences

- Allows a user to define and create new register sequences
- Any of the available register sequences can be selected to get input focus
- The Register Debugger is used to select and add the required registers to the currently active sequence
- If a special non-listed address (e.g. memory) should be accessed a user defined sequence element can be inserted allowing to configure access type, size, mode, ...

Register sequence management implemented

- Register sequences can be loaded and saved independently
- Sequence names can be created and modified
- Individual sequence items can be repositioned or removed
- Sequence items can also be deactivated to prevent from execution
- Values, Masks, Loop counter, ... of the sequence items can be manipulated





Execution of selected register sequences

- Play and Stop buttons available when connection to the target chip is detected
- Special modes and functions are available, depending on the supported chip





Sequence debugging

- It is also possible to debug register sequences
- For this the following debugging functionality is available :
 - Breakpoints
 - 'Single Step'
 - 'Execute to next Breakpoint'
 - 'Stop Sequence'
- Furthermore special sequence items are implemented which support debugging :
 - 'Write' register element
 - 'Read' register element
 - 'Polling Register' elements with a user defined mask and counter
 - 'Write Repeat' element
 - 'Write Repeat Increment' with an address autoincrement

All Elements above are also available User Defined with a definable address and register size



Image Manager - Layout



🔗 Fujitsu GDC Studio						? ×
Eile Project Settings Help						
D 🖪 🗙						±∓□
Register Debugger Stepper Motor Control A/D Converter Beload Timer Unit Sound Generator UART Unit Pulse Generator Chip Control Remote Handler Configuration Fifo Clock Modulator Clock Synthesis	Pixel Format Format32bppRgb Mask Mode Original Organization 32 Bit			Fuj	itsu	
General Purpose IO Sprite Engine Ctrl Sprite Engine SAT Sprite Engine SS Signature Unit Color Lookup Table Dithering Unit Dithering Unit	Information Pixel Data Color Pale Image : File Name File Size Pixel Data Elements Width	Rujitsu.bmp 89846 22448 244 px				(+ (+ ⊝ ⊕
DMA Controller Memory Interface SPI Flash Bun Length Decoder Command Sequencer Register Sequencer panel_nit_ET057003DM6_320x240	Height Bit Depth Pixel Format Bytes Per Line Stride Bytes Per Line Bytes of Image Stride Bytes of Image	92 px 32 bpp Format32bppRgb 976 byte 976 byte 89792 byte 89792 byte	Y			+ OX
spe_test_special_bink_32x240 spe_test_special_bink_32x240 spe_test_special_bink_32x240 spe_test_special_move_320x240 State Sequence Image Manager Memory Editor Memory Dump	· () -	Load Image Close Im	age Save	e As		
					Indigo MB	188F332 🔞 🚲



Image Manager – Pages (1)



Graphics Competence Center

Pixel Format Format4bppIndexed Mask Mode Color Only	Fujitsu	
32 Bit 8 Bit 16 Bit 24 Bit 32 Bit 33 Bit 34 Bit 35 Bit 36 Bit 37 Bit 38 Bit 39 Bit 39 Bit 39 Bit 30 Bit 30 Bit 31 Bit 32 Bit 33 Bit 34 Bit 35 Bit 36 Bit 37 Bit 38 Bit 39 Bit 39 Bit 30 Bit 30 Bit 31 Bit 32 Bit 32 Bit 33 Bit 34 Bit 35 Bit 36 Bit 37 Bit		
OXEREEREE, OXEREREER, OXEREEREE, OXEREEREE, OXEREERE, OXEREER OXEREREE, OXEREREER, OXEREREE, OXEREREE, OXERERE OXEREREE, OXEREREER, OXEREREE, OXEREREE, OXERERE OXEREREER, OXEREREER, OXEREREE, OXEREREER, OXERER OXEREREER, OXEREREER, OXEREREE, OXEREREER, OXERER OXEREREER, OXEREREER, OXEREREE, OXEREREER, OXERER OXEREREERE, OXEREREER, OXEREREER, OXEREREER, OXERER OXEREREER, OXEREREER, OXEREREER, OXEREREER, OXERER OXEREREER, OXEREREER, OXEREREE, OXEREREER, OXERER OXEREREER, OXEREREER, OXEREREE, OXEREREER, OXERER OXEREREER, OXEREREER, OXEREREE, OXEREREER, OXERER OXEREREER, OXEREREER, OXEREREE, OXEREREE, OXERER OXEREREER, OXEREREER, OXEREREE, OXEREREE, OXERERE OXEREREER, OXEREREER, OXEREREE, OXEREREE, OXERERE OXEREREER, OXEREREER, OXEREREE, OXEREREER, OXERER OXEREREER, OXEREREER, OXEREREER, OXEREREE, OXERERE OXEREREER, OXEREREER, OXEREREER, OXEREREER, OXERERE OXEREREE, OXEREREER, OXEREREER, OXEREREER, OXERERE OXEREREE, OXEREREER, OXEREREER, OXEREREER, OXERERE OXEREREER, OXEREREER, OXEREREER, OXEREREER, OXERERE OXEREREERE, OXEREREER, OXEREREERE, OXEREREERE, OXERERE OXEREREERE, OXEREREER, OXEREREERE, OXEREREERE, OXERERE	Pixel Format Format32bppRgb Format32bppRgb Format36bppRgb555 Format16bppRgb555 Format16bppRgb565 Format24bpRgb Format32bppArgb 8 Bit	Fujitsu
	Information Pixel Data Color Palette	1
		OxFFFF0000, OxFFFF2525, OxFFFF3C3C, OxFFFF5151, OxFFFF6363, OxFFFF7474, OxFFFF8585, OxFFFF9494, OxFFFF6363, OxFFFFF6CEC, OxFFFFF700, OxFFFFF8383, OxFFFFFF6CEC, OxFFFFFF900, OxFFFFF86E6, OxFFFFF73F3, OxFFFFFFFF, OxFFFFFFFF





Pixel Format Format4bppIndexed Mask Mode Alpha Only Color Only Alpha Only Inverted Alpha Only Indigo 32 Bit		
Information Pixel Data Color Palette	0x01FF0000, 0x11FF0000, 0x18FF0000, 0x25FF0000, 0x43FF0000, 0x49FF0000, 0x6FFF0000, 0x70FF0000, 0x9CFF0000, 0xA7FF0000, 0xDBFF0000, 0xE0FF0000, 0xEDFF0000, 0xF8FF0000, 0xFEFF0000	•





Extract image information for application development

- Several pages (tabs) provide functions to extract information from an image file and to make this available for a target application
- Automatic source ('C') code generation of image information is possible
- 32 bit organized images can also be saved as binary file (.gdc32dat)
- An 'Information Page' is available to obtain general image information
- A 'Pixel Data Page' is available to extract the core pixel values in a specific organization (8, 16, 24 or 32 bit array)
- A 'Color Palette Page' is available to visualize the colors of indexed format pictures and to create the corresponding Color-Lookup-Tables (CLUTs)

Conversion of pixel data into a chip-specific structure

 Some chips (e.g. Indigo) require a special pixel data arrangement in memory which can be created/converted by this tool





Conversion into different pixel formats

- Images can be read from different image/file formats
- Images can be converted between different pixel formats
- Depending on the input image and the pixel format different 'Mask Modes' will be supported
 - Original
 - Color Only
 - Alpha Only
 - Inverted Alpha Only
 - Alpha + Color





Support for images of the following file types:

- Bmp Bitmap
- Png Portable Network Graphics
- Tiff Tagged Image File Format
- Jpeg Joint Photographic Expert Group
- Gif Graphics Interchange Format

Support for the following pixel formats:

- 1 bpp (bit-per-pixel), indexed
- 4 bpp, indexed
- 8 bpp, indexed
- 16 bpp, different formats
- 24 bpp, RGB
- 32 bpp, ARGB





Support for the following output pixel organization:

- 8 Bit
- 16 Bit
- 32 Bit
- Indigo 32 Bit (Indigo Only)
- Indigo 32 Bit RLD (Indigo Only)
- Ruby 32 Bit ARGB (Ruby Only)
- Ruby 32 Bit ABGR (Ruby Only)
- Ruby 32 Bit RGBA (Ruby Only)
- Any changes are updated immediately in the Picture Box and the corresponding tab pages





🔗 Fujitsu GDC Studio					<u>? ×</u>
Eile Project Settings Help					
🗈 🖬 🗙					¥∓□
Register Debugger Stepper Motor Control A/D Converter Reload Timer Unit I2C Unit Sound Generator UART Unit URE Generator Chip Control Remote Handler Configuration Fifo Clock Modulator Clock Synthesis General Purpose ID Sprite Engine Ctrl Sprite Engine Ctrl Sprite Engine SS Timing Controller Signature Unit Clock Toolker Display Controller Memory Interface SPI Flash Run Length Decoder Register Sequencer Panel_init_ET057003DM6_320x240	Text Fujitsu Rendering Mode Anti Aliasing Anti Aliasing, Grid Clear Type, Grid Bit Per Pixel Bit Per Pixel, Grid Selection Image : Width Height Font : Name Style Underline Strikeout Text Color Background Color Em Height Cell Ascent	164 60 Arial Regular False False Color [Red] Color [White] 48 px 43.45313 px	Fort	Su	
spe_reset_s20x240 spe_test_color_bars_320x240 spe_test_single_color spe_test_single_color spe_test_snecial_blink_32x240	₩ Q-				
Spe_test_special_binin_dc.kero spe_test_special_move_320x240 Sequence Image Manager Memory Editor Memory Editor Memory Dump		Save As			
				Indigo	MB88F332 🙆 ,;





Support for sprite generation during application development

- The Font Manager is used to simply generate attractive letters, numbers or more complex texts
- The following steps are suggested:
 - 1. Select a system font (and a specific character size) (if a special font is needed, copy it to the system font directory first!)
 - 2. Select the required text color
 - 3. Choose the destination image background color
 - 4. Select the required rendering mode
 - 5. Save the resulting output image into a 32 bit (ARGB) file
 - 6. Load the stored file into the Image Converter for further processing

All changes made are updated immediately in the Picture Box





- The following rendering modes are supported:
 - Anti-aliasing
 - Anti-aliasing (Grid)
 - Clear Type (Grid)
 - Bit Per Pixel
 - Bit Per Pixel (Grid)

The output image can be saved in one of the following file formats:

- Bmp Bitmap
- Png Portable Network Graphics
- Tiff Tagged Image File Format
- Jpeg Joint Photographic Expert Group
- Gif Graphics Interchange Format



Memory Editor - Layout



										2 1
										12
: <u>Fi</u> le <u>P</u> roject <u>S</u> ettings <u>H</u> elp										
🖄 🛃 🗙									Ŧ	ŦΟ
🖃 🦢 Register Debugger		0	1	2	3	4	5	6	7	A
Stepper Motor Control	0x00000000	FFFFFFF	FFFFFFF	FFFFFFF	FFFFFFF	FFFFFFF	FFFFFFF	FFFFFFF	FFFFFFF	
- 🌛 A/D Converter	0x00000020	FFFFFFF	FFFFFFF	FFFFFFF	FFFFFFF	FFFFFFF	FFFFFFF	FFFFFFF	FFFFFFF	
	0x00000040	FFFFFFF	FFFFFFF	FFFFFFF	FFFFFFF	FFFFFFF	FFFFFFF	FFFFFFF	FFFFFFF	- P
鑸 12C Unit	0x0000060	00000001	00000010	00000000	00000010	FFFFFFF	FFFFFFF	FFFFFFF	FFFFFFF	
Sound Generator	0x00000080	FFFFFFF	FFFFFFF	00000000	FFFFFFF	02340204	02FF8000	02000100	02FF8000	
UART Unit	0x000000A0	00000200	02FF8000	0004002B	02 FF 8000	03400090	02FF8000	011B0096	02FF8000	
	0x000000C0	FF02FF02	FF02FF02	FF02FF02	FFFFFFF	00000000	3FFF7F00	FFOO1FFF	FFFFFFF	
Chip Control	0x000000E0	FFFFFFF	FFFFFFF	FFFFFFF	FFFFFFF	FFFFFFF	FFFFFFF	FFFFFFF	FFFFFFF	
- A Remote Handler	0x00000100	10200040	32100000	32100000	FFFFFFF	FFFFFFF	FFFF0100	FFFFFFF	FFFF0100	
Configuration Fifo	0x00000120	FFFFFFF	FFFF0100	FFFFFFF	FFFF0100	FFFFFFF	FFFF0100	FFFFFFF	FFFF0100	
- Ulock Modulator	0x00000140	FFFFFFF	FFFF0100	FFFFFFF	FFFF0100	FFFFFFF	FFFF0100	FFFFFFF	FFFF0100	
Clock Synthesis	0x00000160	FFFFFFF	FFFF0100	FFFFFFF	FFFF0100	FFFFFFF	FFFFFFF	FFFFFFF	FFFFFFF	
General Purpose IU	0x00000180	FFFFFFF	FFFFFFF	FFFFFFF	FFFFFFF	FFFFFFF	FFFFFFF	102030FF	20060000	
Sprite Engine Utri	0x000001A0	00000000	00000045	102C0000	FFFFFFF	FFFF1106	FFFF0000	FFFF8126	FFFF0000	
Sprite Engine SAT	0x000001C0	FFFF0064	FFFF0000	FFFF0D20	FFFF0000	FFFF0E41	FFFF0000	FFFF5030	FFFF0000	
🥔 Sprite Engine SS	0x000001E0	FFFF0004	FFFF0000	FFFF60A0	FFFF0000	FFFFFFF	FFFFFFF	FFFFFFF	FFFFFFF	
Iming Controller	0x00000200	FFFFFFF	FFFFFFF	FFFFFFF	FFFFFFF					
Signature Unit										
Difference Llag										
Ditnering Unit										
Display Controller DMA Controller										-
Manayu Interface										-
CPI Elash										
Pun Longth Decoder										
Command Sequencer										-
Begister Sequencer										- X
papel init ET057003DM6 320v240										-
pane_mic_100100000mic_020x240 we reset 320x240										-
spe_test_color_bars_320x240	: 🛣 🗥 -									
spe_test_single_color	SALK, She									
→→ spe_test_special_blink_32x240			- Offeet (Hev)	Item Cou		-Val	ua (Hav) —	_		
→→ spe test special move 320x240			Unsec(nex)				ue (rrex)			
→→ Example Sequence			0		32		102030FF			_
- 🗀 Image Manager				[E	nter the new Valu	ue (Hexadecimal) fo	r the selected Iter	n
- 🔁 Font Manager							nange : Edit or + 	r, - Key(s) r		
Memory Editor				1			eave : Press Esca	ape or Leave Contro	bl	
Memory Dump			Read Memory B	lock Write Me	mory Block					_
							Γ	Indiao	MB88E332	. 🔊



Flash Editor - Layout



Fujitsu GDC Studio ? × File Project Settings Help 🖄 🛃 🗙 ±∓□ 🖃 🗁 Register Debugger 0 1 2 3 4 5 6 - 🌙 Stepper Motor Control 0x000A0000 12345678 FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFF - 🌙 A/D Converter 0x000A0020 FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFF 🌛 Reload Timer Unit 0x000A0040 FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFF - 실 12C Unit 0x000A0060 FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFF Sound Generator 0x000A0080 FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFF - 🎒 UART Unit 0x000A00A0 FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFF 🌛 Pulse Generator 0x000A00C0 FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFF 🕘 Chip Control 0x000A00E0 FFFFFFF FFFFFFF FFFFFFF 87654321 FFFFFFF FFFFFFF FFFFFFF FFFFFFF 🍓 Remote Handler 0x000A0100 FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFF Configuration Fifo 0x000A0120 FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFF 🍓 Clock Modulator 0x000A0140 **** **** ***** ***** ***** **** **** 🌛 Clock Synthesis 0x000A0160 FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFF 🍓 General Purpose IO 0×000&0180 ******* FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFF ****** ****** - 🌙 Sprite Engine Ctrl 0x000A01A0 FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFF Sprite Engine SAT. 0x000A01C0 FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFF ****** 🌛 Sprite Engine SS 0x000A01E0 FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFF ***** 🌛 Timing Controller 0x000A0200 FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFF ⇒ 🍓 Signature Unit 0x000A0220 FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFF 🎯 Color Lookup Table 0x000A0240 FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFF FFFFFFF 🍓 Dithering Unit 0x000A0260 FFFFFFF FFFFFFF Display Controller DMA Controller -- 🌛 Memory Interface 🌛 SPI Flash J 🕘 Run Length Decoder 🎒 Command Sequence 🖂 🗁 Register Seguencer Erase Flash Chip panel_init_ET057003DM6_320x240 Erase Single Flash Sector None www.spe_reset_320x240 None Offset (Hex) Value (Hex) 87654321 → Example Sequence 🚞 Image Manager 🛅 Font Manager 📜 Flash Editor Read Memory Block Flash Memory Block 📜 Flash Dump Indigo MB88F332 Scanning Flash Memory Block for Content Detected * PROCESSING FINISHED with SUCCESS .€





Useful for several tasks

- Supports debugging and validation for both hardware and software,
 e.g. by dumping the memory content to check the proper loading of sprites
- Manipulation of data stored in either RAM or flash memory by:
 - reading/writing single memory cells (4 bytes each) or
 - reading/writing complete memory blocks from/to a specified address
 - Limited to 512 items per block

Using the editor is straightforward:

- Enter the start address (of register, RAM or flash memory)
- Insert the required number of items to be read
 (an item is the smallest unit to be dumped = 4 bytes width)
- Read out the memory block into virtual memory by pressing the corresponding button
- Manipulate single items
- Write the manipulated virtual memory block (or parts of it) back to the start address or any other specified address



Memory Dump - Layout



🖓 Fujitsu GDC Studio	
Eile Project Settings Help	
🖄 🖪 🗙	
Register Debugger Septem Koto Control A/D Converter A/D Converter Sound Generator Control Spite Generator Control Spite Engine SAT Sp	
Image Manager Offset (Hex) Item Count	
Font Manager Memory Editor Memory Dump Read Memory Block Write Memory Block Load Dump from Save Dump to	
Indigo M888F332	2 .:





🔗 Fujitsu GDC Studio ? × File Project Settings Help 🖄 🔒 🗙 ±∓□ E 🗁 Register Debugger Hex Dump 0 Stepper Motor Control A/D Converter 0x02010000 0x00014000 0x00000002 0x02010000 0x00014004 0x00130200 0x02010000 0x00034004 0x0198000(٠ 🍓 Reload Timer Unit 0x00034010 0x01040000 0x02010000 0x00034014 0x00F000F4 0x02010000 0x00034000 0x80060004 0x0201000(🍓 12C Unit 0x00000000 0x02010000 0x00034204 0x00000100 0x02010000 0x00020024 0x00000140 0x02010000 0x0002852(🍓 Sound Generator 0x02010000 0x0002853C Save Binary File ? × 🍓 UART Unit 0x00028550 0x00000090 0x00100091 0xFFFFFFFF Pulse Generator Save in: 🛅 binary 🔻 Ġ 🥼 📂 🖽 -0 Chip Control 🍓 Remote Handler Ì -Configuration Fifo 🌛 Clock Modulator My Recent Documents -🍓 Clock Synthesis 🍓 General Purpose IO B 🌙 Sprite Engine Ctrl 🍓 Sprite Engine SAT C 🍓 Sprite Engine SS Timing Controller ٠ 🌙 Signature Unit 🌛 Color Lookup Table * dy Docume 🌙 Dithering Unit 🌛 Display Controller 🌛 DMA Controller ÷ Memory Interface dy Compute 🍓 SPI Flash I 🌛 Run Length Decoder Command Sequencer 1 E 🗁 Register Sequencer My Networ Places FlashExampleDump -Save File name: →→ spe_reset_320x240 • Cancel Save as type: Binary File (*.gdc32dat) In spe_test_color_bars_320x240 \bigcirc spe_test_single_color spe_test_special_blink_32x240 Offset (Hex) Item Count spe_test_special_move_320x240 0 82 → Example Sequence 🔁 Image Manager 🔁 Font Manager 📃 Flash Editor Read Memory Block Flash Memory Block Flash Dump Indigo MB88F332





Supported Features :

- Provides an overview of large memory blocks, i.e. many more items can be dumped and listed on the screen than in the Memory Editor (currently limited to 8MByte)
- Reads register, memory or flash areas
- Saves the dumped data into binary files (.gdc32dat)
- Reloads binary files into the dump view also converted images
- Loads Register Sequencer files (interpreted as command list values on Indigo)
- Writes the dump view content to a specified address in the register, memory or flash area
- An internal state machine is implemented to optimize data writing into flash memory (e.g. check if empty, check for new content before writing, automatic merges of new data and already existing identical data based on a read and comparison of corresponding sectors, ...)





GDC Studio runs on Microsoft operating systems

- Windows 2000
- Windows XP
- Windows Vista (when supported from the interface driver)
- GDC Studio's Setup Wizard provides an automatic installation process

Interface between a PC/Laptop and the MB88F332 (Indigo):

SPI

- USB to SPI converter box 'Aardvark I2S/SPI Host Adapter'
- GDC Studio uses the USB driver provided with the 'Aardvark' hardware
- Must be purchased directly from the company 'TOTAL PHASE'
- http://www.totalphase.com/products/aardvark_i2cspi/
- Hardware is not delivered with the tool and is not included in the license fee





Interface between a PC/Laptop and the MB86298 (Ruby):

SPI

- USB to SPI converter box 'Aardvark I2S/SPI Host Adapter'
- GDC Studio uses the USB driver provided with the 'Aardvark' hardware
- Must be purchased directly from the company 'TOTAL PHASE'
- http://www.totalphase.com/products/aardvark_i2cspi/
- Hardware is not delivered with the tool and is not included in the license fee

PCle

• For this chip design the Fujitsu GD Studio supports the PCIe connection of the Ruby Evaluation Board directly





- An installation must be licensed for every PC workplace (machine)
- Licenses for multiple workplaces (groups) are also possible
- A free tool is provided which supports the authorization procedure that extracts the required information from the PC workplace hardware:
 - → 'Fujitsu GDC Studio Authorization Support Tool'
- This information can be selected and sent to Fujitsu GCC to get an authorization for the use of Fujitsu GDC Studio
- The license fee depends on # installations/PC workplaces:
 - Licenses are available for groups of:
 - up to 4 workplaces up to 6 workplaces
 - up to 8 workplaces
 - up to 10 workplaces
 - up to 12 workplaces
 - up to 14 workplaces
 - up to 16 workplaces



THE POSSIBILITIES ARE INFINITE