Mi-V RV32 簡易チュートリアル

PolarFire SoC Discovery Kit 版



目次

1.	. 概要	4
	1-1. 本資料について	4
	1-2. Mi-V コア	4
2.	. SoftConsole のインストール (未インストールの場合)	5
3.	. ハードウェア	6
	3-1. Libero SoC プロジェクト作成	6
	3-2. SmartDesign の新規作成	9
	3-3. Mi-V RV32	11
	3-4. Clock Conditioning Circuitry (CCC)	15
	3-5. PolarFire Initialization Monitor	18
	3-6. CoreReset_PF	20
	3-7. CoreJTAGDebug	22
	3-8. MIV_ESS	23
	3-9. ブロックの接続	26
	3-10. Drive Constraints	38
	3-11. 論理合成	39
	3-12. ピンアサイン	39
	3-13. 配置配線	41
	3-14. 書き込み	42
4.	. ソフトウェア	43
	4-1. SoftConsole 起動	43
	4-2. Driver の入手	45
	4-3. 新規プロジェクト作成	48
	4-4. Driver のインポート	51
	4-5. プロパティ設定	54
	4-6. ソースコードの作成	62
	4-7. リンカスクリプトの編集	66
	4-8. fpga_design_config.h の用意	68
	4-9. #include パスの修正	71
	4-10. ビルド	72
5.	. 実行	72
6.	. ソフトウェアをデバイスへ書き込んでみよう	75
7.	. UART を動かしてみよう	80
	7-1. ハードウェア	80



	7-2. ソフトウェ	7	 6
8.	関連ドキュメン	Ի	 3



1. 概要

1-1. 本資料について

Mi-V(Microchip 社のソフト CPU)をはじめて使用される方を対象とした 簡易チュートリアルです。主に Libero SoC での新規プロジェクト作成から Mi-V でプログラムを作成し LED を点滅させるまでの手順を記載します。 Libero SoC の使い方の説明は割愛しています。

1-2. Mi-V コア

Mi-V のコアには種類があります。本資料では最新コアである Mi-V RV32 を 使用します。

We offer a comprehensive ecosystem of software tool chains and IP cores for FPGA designs. Mi-V RV32 RISC-V soft CPUs are available for PolarFire [®] SoC, PolarFire [®] , RTG4 [™] , SmartFusion [®] 2 and IGLOO [®] 2 FPGAs with complete design support through Libero [®] SoC Design Suite. The Eclipse-based SoftConsole IDE contains a development environment, a Renode emulation platform, GCC complete and debugger needed for C/C++ embedded firmware development.								
The Libero SoC Design Suite and SoftConsole development environments provide all the required tools to integrate Mi-V soft CPUs in our FPGAs and develop, test and debug embedded firmware.								
RISC-V Soft CPU Mi-V RV32IMA_L1_AHB Mi-V RV32IMA_L1_AHB Mi-V RV32IMA_L1_AXI								

LEs	4k-20k	26k	10k	10k
CoreMark [®] Score	0.18-2.48	2.01	2.01	2.01
Cache Size	1 KB I\$	2 KB I\$, 8 KB D\$	2 KB I\$, 8 KB D\$	2 KB I\$, 8 KB D\$
Tinklis Counted Manness (TCM)	Yes, Configurable Depth	ni / n	81 <i>7</i> .8	51 / A

1-3. 使用開発キット

PolarFire SoC Discovery Kit

※ペリフェラルは少ないですが比較的小さくて持ち運びやすい

お手頃な開発キットです。

※今回ハード CPU の部分である MSS(Microprocessor Sub-System)は使用しません。

1-4. 動作確認バージョン

Libero SoC v2024.2 SoftConsole v2022.2



1-5. 動作させるデザイン

PolarFire SoC Discovery Kit の LED1、LED2、LED3、LED4 を 0101、1010 と交互に点灯、消灯させます。

構成:



- 2. SoftConsole のインストール (未インストールの場合)
 - ・Mi-Vのソフトウェアは SoftConsole にて開発します。
 - ・未インストールの場合は事前にダウンロード、インストールします。
 - SoftConsoleのバージョンは LiberoSoC のバージョンと連動していません。
 2025 年 2 月時点では v2022.2 が最新です。
 - ·SoftConsole はライセンス不要で使用可能です。

ダウンロードリンク:

https://www.microchip.com/en-us/products/fpgas-and-plds/fpga-and-soc-design-tools/soc-fpga/softconsole



3. ハードウェア

- 3-1. Libero SoC プロジェクト作成
- ① Libero SoC を起動します。



② スタート・ページ > "New"

🕘 Libero



もしくは Project > New Project

۷	Liber	D						
Pro	oject	File	Edit	View	Design	Tools	Help	
	New	Proj	ect		Ct	rl+N		
Ê	Oper	n Pro	ject		Ct	rl+O	_	
×	Close	е						
	Save				Ct	rl+Shift	:+S	oero So
	Savo	Ac			C+	d i Shifi		

をクリックし、New Project ウィザードを開きます。



③ Project nam、Project location を設定し Next > を押します。
 お手元の PC のご都合の良いフォルダへ設定して下さい。
 Project name: miv_top

Project location:	C:¥miv_lab

🕑 New project						-		×
Project details Specify project deta	ils							
Project Details	Project name:	miv_top						
Device Selection	Project location:	C:/miv_lab					Browse	
Device Settings	Description:							
Design Template								
Add HDL Sources	Preferred HDL type Enable block cm Block flow enables It could include tim	:: Verilog <u> </u>	usable component that car vsical constraints, placeme	n be instantiated int or routing.	into another des	sign. A block co	omponent r	nay r
Add Constraints			,,-					
Help				< Back	Next >	Finish	Can	;el

(4) Device selection $\kappa \tau$

Family: PolarFireSoC を選択 Search part: MPFS095T-1FCSG325E を入力 MPFS095T-1FCSG325E を選択し Next> を押します。 🕑 New project _ \times Device selection Select a part for your project from the part number list Selected part: MPFS095T-1 FCSG325E Part filter Project Details Family: PolarFireSoC Die: All ➡ Package: All • Speed: All Range: All -**Device Selection** Reset filters Device Settings Search part: MPFS095T-1FCSG325E LSRAM Part Number DFF User I/Os uSRAM Math Add HDL Sources Add Constraints Libero • Help < Back Next > Finish Cancel



⑤ Device settings ページにてデフォルトのまま Next>を押します。

O New project		-		×
Device settings Choose device settings for your project	Selected pa	art: MPFS09	5T-1 FCS	G325E
Project Details Core Voltage : 1.0				
Default I/O technology: Default I/O Editor to C	change individ	lual I/O attri	butes.	
Device Settings When this option is enabled, please ensu Initialization Monitor IP Core and Tampe Gonfigured to "Latch System controller suspended mode UK EDOM DOCOC 1 SOME	ure that the ar IP Core are outputs [®] and to the	Ithe		
Add HDL Sources Add Constraints If System controller suspended mode is enabled, the following operations will not be a Add Constraints If System controller suspended mode is enabled, the following operations will not be a Add Constraints If System controller suspended mode is enabled, the following operations will not be a Add Constraints If System controller suspended mode is enabled, the following operations will not be a Add Constraints If System controller suspended mode is enabled, the following operations will not be a Device reset and device zerolation Tamper responses SPI-Master Im-Application Programming (IAP) For further information, refer to the System Services section in the PolarFire FPGA	available: A Security Use	er Guide (UG	60753).	
Help Sack N	lext >	Finish	Canc	;el

⑥ Add HDL source files にてデフォルトのまま Next> を押します。

ONew project			_	□ ×
Add HDL source files Specify HDL files to im	port/link to your project.		Selected part: MPFS095	T-1 FCSG325E
Project Details	Import file Link file			Delete
Device Selection	File type	File name	File location	
Device Settings				
Add HDL Sources				
Help		< Back	Next > Finish	Cancel



⑦ Add constraints にてデフォルトのまま Finish を押します。

🕑 New project			-		\times
Add constraints Specify constraint files f	for timing or physical constraints.		Selected part: MPFS09	5T-1 FCSC	3325E
Project Details	Import file Link file			Delete	
Device Selection	File type	File name	File location		
Device Settings					
Add HDL Sources					
Add Constraints					
Libero					
Help		< Back	Next > Finish	Cance	el

- 3-2. SmartDesign の新規作成
- 本トレーニングでは回路図エディタにてデザインを作成します。 Design Flow 内にて Create SmartDesign をダブルクリック。

Design Flow							8 ×
0	Please	select a root		Ð	0	мá	🗿 🜮
Tool							
	Create Design						
	🐴 Import MSS						
	Create SmartDes	ign					
	📋 Create HDL						
	🖓 🔐 Create SmartDe						
-	📋 Create HDL Testb	ench					
- B-	Verify Pre-Synth	nesized Design					
	Simulate						
	Constraints						
	🗟 Manage Constrai	ints					
÷	Implement Design						
	🖓 Open Netlist Vie	wer					
	Synthesize						
	Verify Post-Synt	hesized Design					
	• Generate Sin	nulation File					
Simulate							
Configure Register Lock Bits							
- 🔐 Place and Route							
🖻 Edit Post Layout Design							
	Verifv Post Lavo	ut Implementation					•
Design Flow	Design Hierarchy	Stimulus Hierarchy	Catalog	Com	ponen	ts	Files



② 名前を入力し OK を押します。

Name: miv_top

Create New SmartDesi	gn ?	×
Name:		
miv_top		
Help OK	Car	ncel

③ Design Hierarchy タブを開き、作成した SmartDesign を右クリック、
 Set As Root を選択します。

Design Hierarchy		₽×	
0	Please select a root		
A Build Hierarchy	🔍 Show: Components 💌 🗉 🖿	? 🗞	
O O work O O O	Set As Poot		
E Components	Open Component		
	Generate Component		
	Export Component Description(Tcl)		
	Hierarchical Export Component Description(Tcl)		
	Rename Component		
	Create I/O Constraint from Module		
	Create Testbench		
	Delete		
	Copy File Path		
	Show Module Parameters		
	Export Parameter Report		
Design Flow Design Hierarchy	Properties	Files	
Log	Show Module		



3-3. Mi-V RV32

まずソフト CPU Mi-V のプロセッサーコアを置きます。
 Catalog タブを開き Processors 内の Mi-V RV32 を

Smart Design 上へドラッグ&ドロップします。

Uibero - C:¥miv_lab¥miv_top¥miv_top.prjx*					_		×
Project File Edit View Design Tools SmartDesign H	elp						
] 🗅 🚔 🔛 🗠 🝋 🖌 🙆 🛛 🚰							
Catalog		Ξ×	SD miv_top* 🗗 🗙	Reports & ×	StartPage	8×	Ŧ
IP Catalog	-		🔂 🗸 🗊 🖸			D D	D "
mi-v Q	 Simulation Mode 	<u>()</u> –					
Name /	Version						
Processors	,						
Mi-V RV32	3.1.200						
MI-V RV32	3.1.100						
	3.0.100						
	2.1.100						
	2.3.100						
	2.1.100		>				
	2.1.100						
, Documentation:							
MIV RV32 v3.1 User Guide.pdf							
DS00003723B MIV RV32 Migration Guide off							
riscv-debug-release v0.13.2.pdf		•					
		_					-
Design F··· Design Hierar··· Stimulus Hierar···	Cat… Compone…	F	•				•

Create Component ウィンドウが立ち上がります。
 そのまま OK を押します。

Create Component	?	\times
Name:		
MIV_RV32_CO		
HelpOK	Car	ncel



③ Mi-Vのプロセッサーコアを設定します。今回は下記のように設定します。

Interface Options

AHB Initiator : None

APB Initiator : APB3、 AHB Mirrored I/F $\sim f \pm y \gamma$

AXI Initiator : None

Tightly-Coupled Memory (TCM) Options

TCM:チェックを入れる

Timer Options

Internal MTIME: チェックを外す

Internal MTIME IRQ: チェックを外す

Configurator	-		×
Mi-V RV32 Configurator			
Microsemi:MiV:MIV_RV32:3.1.200			
Configuration Memory Map			
Extension Options			
C: 🔽 F: 🗌 M: 🔽 Multiplier: Fabric 🗾 🚯			
Therface Options			
AHB Initiator: None 💌 AHB Mirrored I/F: 🔽 🕄			
AXI Initiator: None 💽 AXI Mirrored I/F: 🚺			
ICACHE: 🔽 🚯 Multi-Interface IM: 🗖 🚯			
Reset Vector Address			
Upper 1 6 bits (Hex): 0x8000 Lower 1 6 bits (Hex): 0x0			
BootROM Option			
BootROM: 🔲 🚯 Reconfigable: 🥅 🚯			
Tightly Coupled Memory (TCM) Options			
TCM: 🔽 🕄 TCM Access Support (TAS): 🗌 🚯			
Interrupt Options			
External System IRQs: 🛛 🗾 🚯			
Vectored Interrupts: 🔲 🚯			
Timer Options			
Internal MTIME: MTIME Prescaler: 100			
Debug: IV 🚺 Trace Interface: 🚺 Hart ID: 0x0 🚺			
Performance and Reliability Options			
Register Forwarding: 🗌 🕄 ECC Enable: 📄 🕄 Disable MACCs: 🔲 🕄			
GPR Registers: 🗌 🚯 TCM Registers: 🗐 🚯 ICACHE Registers: 🗐 🚯			
Help 🔻	ОК	Can	cel



補足: Interface Options:
Mi-V RV32 プロセッサと GPIO を繋ぐため AMBA APB(Advanced Peripheral Bus)
を有効にしています。
┌ Interface Options
AHB Initiator: None 🖃 🛃 AHB Mirrored I/F: 🧮 🚯
APB Initiator: 🗚 B3 💽 🔥 APB Mirrored I/F: 🗌 🚯
AXI Initiator: None 🖃 🕭 AXI Mirrored I/F: 🔲 🚯
ICACHE: Multi-Interface IM:

補足: APB Mirrored I/F について:				
今回チェックを入れている APB Mirrored I/F についての説明は、				
青いiマークにカーソルをあてると確認可能です。				
"This feature can bw used to directly connect a single target component (e.g. RAM)				
without the need of a bridge"				
Interface Options				
AHB Initiator: None 💽 AHB Mirrored I/F: 🔽 🚯				
APB Initiator: APB3 💽 APB Mirrored I/F: 🔽 🖲				
AXI Initiator: None 💽 AXI Mirrored I/F: 🔽 GThis feature can be used to directly connect a single target component (e.g. RAM) without the need of a bridge				

補足: Tightly-Coupled Memory (TCM) Options: TCM は Tightly-Coupled Memory (密接合メモリ)の略になります。 本チュートリアルでは一例として TCM のアドレスを Mi-V の RESET_VECTOR_ADDRESS に設定します。 引用「The processor can be booted from this memory region by setting the RESET_VECTOR_ADDRESS to the address of the TCM.」 https://ww1.microchip.com/downloads/aemDocuments/documents/FPGA/ProductDocu ments/UserGuides/ip_cores/directcores/MIVRV32v31UserGuide.pdf#page=15 Tightly Coupled Memory (TCM) Options TCM: ▼ ▲ TCM Access Support (TAS): □ ①



④ アドレスを変更します。

Reset vector はデフォルトでは 0x8000_0000 となっています。

Reset Vector Address		
Upper 16 bits (Hex): 0x8000	🔥 Lower 16 bits (Hex): 0x0	

現在 0x8000_0000 がアドレス範囲外との警告が表示されているため変更対応します。

Memory Map タブを開き、

TCM Address の Upper 16bits (Hex)を 0x4000 から 0x8000 へ変更し OK を押します。

Reset Vector Address (RVA) must be set to an address in the valid address range. The valid address range is any address within the active AHB, APB, AXI or TCM address ranges

Configurator	- 0	×
Mi-V RV32 Configurator		
Microsemi:MiV:MIV_RV32:3.1.200		
Configuration Memory Map		
AHB Initiator Address		
Start Address: Upper 16 bits (Hex): 0x8000	Lower 16 bits (Hex): 0x0	
End Address: Upper 16bits (Hex): 0x8fff	Lower 16 bits (Hex): Oxffff	
APB Initiator Address		
Start Address: Upper 16 bits (Hex): 0x7000	Lower 16 bits (Hex): 0x0	
End Address: Upper 16bits (Hex): 0x7fff	Lower 16 bits (Hex): 0xffff	
AXI Initiator Address		1
Start Address: Upper 16 bits (Hex): 0x6000	Lower 16 bits (Hex): 0x0	
End Address: Upper 16bits (Hex): 0x6fff	Lower 16bits (Hex): 0xffff	
TCM Address		
Start Address: Upper 16 bits (Hex): 0x8000	Lower 16 bits (Hex): 0x0	
End Address: Upper 16bits (Hex): 0x8000	Lower 16bits (Hex): 0x3fff	
- TCM Access Support (TAS) Address		
Start Address: Upper 16bits (Hex): 0x4000	Lower 16bits (Hex): 0x0	
End Address: Upper 16 bits (Hex): 0x4000	Lower 16bits (Hex): 0x3fff	
BootROM Address		1
Source Start Address: Upper 16bits (Hex): 0x8000	Lower 16bits (Hex): 0x0	
Source End Address: Upper 16 bits (Hex): 0x8000	Lower 16bits (Hex): 0x3fff	
Destination Address: Upper 16bits (Hex): 0x4000	Lower 16bits (Hex): 0x0	
Help 🔹	OK	L



OK 押下後、Smart DesignMi-V RV32 コアのブロックが Smart Design 上に 表示されます。



- 3-4. Clock Conditioning Circuitry (CCC)
- Catalog タブより CCC を選択し、SmartDesign 上へドラッグ&ドロップします。 いわゆる PLL の IP になります。

IP Catalog			•	Simulation Mode	<u>o</u> -
Name			Version	1	
Clock &	/Janagement				
Clock	Conditioning Circuit	y (CCC)	2.2.220		
Clock	Conditioning Circuit	y (CCC)	2.1.104		
⊨ I/O					
Polari	Fire IOD CDR Clocking	9	2.1.111		
Polari	Fire IOD Generic Trans	smit Interfaces Clocking	1.0.128		
PolarFire	Features				
Clock	Conditioning Circuit	y (CCC)	2.2.220		
Clock	Conditioning Circuit	y (CCC)	2.1.104		
Polari	Fire IOD CDR Clocking	9	2.1.111		
Polari	Fire IOD Generic Trans	smit Interfaces Clocking	1.0.128		
No core selec	ted				
Docign Flow	Decign Hierarchy	Stimulus Historsby	Ootolog	Componente	Files
Design Flow			Catalog	Components	



② Create Component ウィンドウにてデフォルトのまま OK を押します。

Create Component	?	\times
Name:		
PF_CCC_C0		
HelpOK	Car	ncel

3 Input Frequency $\mathit{k\tau}$

Input Frequency : 50MHz

Bandwidth : High

に設定し	まう	す。
------	----	----

Configurator	- 🗆 ×
Clock Conditioning Circuitry (CCC)	
Actel:SgCore:PF_CCC:2.2.220	
Configuration PLL-Single	_ <u>*</u>
Clock Options PLL Output Clocks	-
Dinput Frequency	
Input Frequency 50 MHz 🗆 Backup Clock Bandwidth High 💌 = 1.667 MHz	
 Delay Line 	
Enable Delay Line	_
Reference Clock Delay C Feedback Clock Delay Delay Steps: 1	
Power / Jitter	-PLL_POWERDOWN_N_0 PLL_LOCK_0-
 Maximize VCO for Lowest Jitter VCO = 4800 MHz Minimize VCO for Lowest Power 	PF_CCC
E Feedback Mode	
Post-VCO 🔄 🔽 Reset Outputs On PLL Lock	
E Features	
🗌 Integer Mode 🚯	
SSCG Modulation	
 Enable Dynamic Reconfiguration Interface (DRI) Expose PowerDown Port 	
Log	
Messages SErrors A Warnings	
Awarning: If you select Post - VCO as a Feedbar PLL Lock' is enabled outputs will be resynchroni locks, but will not be resynchronized with the P	ck Mode, and if 'Reset Outputs on zed between each other after the PLL LL reference clock.
Help -	OK Cancel



補足: CCC の Bandwidth 設定:
Bandwidth はリファレンスクロックの品質に応じて設定します。
引用「If the reference clock has a significant amount of jitter, use lower bandwidth
to filter the noise. If a higher quality reference clock is used, fast lock time is
achieved by using a higher bandwidth value.]
https://ww1.microchip.com/downloads/aemDocuments/documents/FPGA/ProductDocu
$\underline{ments/UserGuides/Microchip_PolarFire_FPGA_and_PolarFire_SoC_FPGA_Clocking_Res}$
ources_User_Guide_VB.pdf#page=33
Clock Options PLL Output Clocks
Input Frequency
Input Frequency 50 MHz 🗌 Backup Clock
Bandwidth High = 0.893 MHz

④ Output Clocks タブを開き、Output Clock 0 の

Requested Frequency を 83.333MHz に設定し、OK を押します。

Configurator	
Clock Conditioning Circuitry (CCC)	
ActelSeCore: PE CCC 2 2 220	
	1
Configuration PLL-Single	
Clock Options PLL Output Clocks	
For best results, put the highest frequency first.	
Output Clock 0	
Requested Frequency (83.333) MHz C Actual Lower 83.333 MHz C Actual Higher 83.333 MHz PLLPOWERDOWNNO PLLLOCKO-	
Requested Phase 0 Degrees C Actual Lower 0 Degrees C Actual Higher 0 Degrees PF_CCC	
Dynamic Phase Shifting Expose Enable Port	
I Global Clock Global Clock (Gated) HS I/O Clock Dedicated Clock	
Output Clock 1	
Enabled	
Log	
EMessages SErrors A Warnings O Info	
Avaining: If you select Post - VCO as a Feedback Mode, and if 'Reset Outputs on PLL Lock' is enabled outputs will be resynchronized between each other after the PLL locks, but will not be resynchronized with the PLL reference clock.	
Help Cancel OK Cancel	



⑤ Warning ウィンドウが出るため OK を押します。



⑥ 同様に Information ウィンドウにて OK を押します。

Information	×
Design "PF_CCC_C0" was successfully generate	ed.
ОК	

- 3-5. PolarFire Initialization Monitor
- ① PolarFireSoC Initialization Monitor を追加します。

P Catalog	📃 🍳 🔻 🔲 Simulation Mod	le 🔞 🔻
Name		
En Clock & Management		
PolarFireSoC Initialization Monitor	1.0.307	
Memory & Controllers		
ScoreSDR_AHB	4.4.107	
Peripherals		
🖙 🍉 Core1553BRM	4.3.104	
🔤 🍉 CoreSDR	4.3.103	
🖙 CoreSDR_AHB	4.4.107	
🖙 🍉 CoreSGMII	3.3.101	
- SGMII	3.2.101	
- CoreSGMII	3.1.102	
PolarFireSoC Features		
PolarFireSoC Initialization Monitor	1.0.307	
Processors		-
No core selected		
esign Flow Design Hierarchy Stimulus Hier	archy Catalog Components	Files





② Create Component ウィンドウにて OK を押します。

Create Component	?	\times
Name:		
PFSOC_INIT_MONITOR_C	ol	
Help OK	C	ancel

③ 下記のように設定しOKを押します。

Enable Bank0 calibration status pin (BANK_0_CALIB_STATUS): チェックを外す Enable Bank1 calibration status pin (BANK_1_CALIB_STATUS): チェックを外す Enable Bank0 VDDI status pin (BANK_0_VDDI_STATUS): チェックを入れる ※ 今回は Bank 0 に繋がっている LED を使用します。

Configurator	- 🗆 X
PolarFireSoC Initialization Monitor Configurat	tor
Microsemi:SgCore:PFSOC_INIT_MONITOR:1.0.307	
Bank Monitor Dynamic Recalibration Simulation Options A Calibration Monitor Enable Bank0 calibration status pin (BANK_0_CALIB_STATUS) Enable Bank1 calibration status pin (BANK_1_CALIB_STATUS)	
VDDI Monitor Enable Bank0 VDDI status pin (BANK_0_VDDI_STATUS)	SRAM_INIT_DONE
Enable Bank1 VDDI status pin (BANK_1_VDDI_STATUS)	USRAM_INIT_FROM_SNVM_DONE - USRAM_INIT_FROM_UPROM_DONE - USRAM_INIT_FROM_SPIDONE -
Latch System Controller outputs 🗌	SRAM_INIT_FROM_SNVM_DONE SRAM_INIT_FROM_UPROM_DONE SRAM_INIT_FROM_SPILDONE AUTOCALB_DONE PFSOC_INIT_MONITOR
Help -	OK Cancel



補足: DDR メモリを使用する場合

DDR メモリを使用する場合 BANK_#_CALIB_STATUS 信号を有効にします。

引用「Note: IOs must be calibrated before initiating the training logic of the DDR controller. This requires generating a reset signal by ANDing the DEVICE_INIT_DONE and BANK_#_CALIB_STATUS signals of the PFSOC_INIT_MONITOR IP. BANK_# refers to the BANK where DDR subsystem is placed.」

https://ww1.microchip.com/downloads/aemDocuments/documents/FPGA/ProductDocu ments/UserGuides/microchip_polarfire_fpga_and_polarfire_soc_fpga_power_up_and_reset _user_guide_vb.pdf#page=14

3-6. CoreReset_PF

① CoreReset_PF を SmartDesign 上へドラッグ&ドロップします。

Catalog	8 ×
IP Catalog	
reset	🔍 👻 🗌 Simulation Mode 🍥 🔻
Name	🛆 Version
Peripherals	
CoreReset_PF	2.3.100
CoreReset_PF	2.2.107
CoreReset_PF	2.1.100
Solutions-MotorControl	
🖙 🗫 Rate Limiter	4.2.0
No core selected	
1	
Design Flow Design Hierarchy Stimulus	s Hierarchy Catalog Components Files

② Create Component ウィンドウにて OK を押します。

Create Comp	onent	?	\times
Name:			
CORERESET_PF	F_CO		
Help	OK	Car	ncel



③ CoreReset_PF IP の設定画面にてデフォルトのまま OK を押します。

Configurator		-		×
CoreReset_	PF C	onfig	urato	or
Actel:DirectCore:C	ORERES	ET_PF:2.3	8.100	
Configuration				
Testbench: User _	•			
Help 🔻		OK	Car	icel

補足: PolarFire 以外のデバイスを使用する場合

CoreReset_PF は PolarFire 用の IP です。 他のデバイスを使用する場合は reset_synchronizer を HDL で用意し、

SmartDesign 上へドラッグ&ドロップします。

サンプルソースは MIV_RV32 User Guide をご参照ください。

引用「Note: The difference between the PolarFire and the other design is the use of the PolarFire Initialization Monitor and CoreReset_PF. The HDL reset synchronizer acts as the CoreReset_PF, as this IP is only available on PolarFire. For other families, the reset synchronizer is used. The HDL code for the reset synchronizer is available in section 6.3.2 RTG4 (IG2/SF2) RESETN.」

https://ww1.microchip.com/downloads/aemDocuments/documents/FPGA/ProductDocu ments/SupportingCollateral/MIV_RV32+v3.1+User+Guide.pdf



3-7. CoreJTAGDebug

① CoreJTAGDebug を SmartDesign 上にドラッグ&ドロップします。

Catalog			Β×
IP Catalog			
jtag	■	🔲 Simulation Mode	0 -
Name	Δ	Version	
🖶 Macro Library			
UJTAG		1.0	
UJTAG_SEC		1.0	
Processors			
CoreJTAGDebug		4.0.100	
CoreJTAGDebug		3.1.100	
CoreJTAGDebug		3.0.100	
CoreJTAGDebug		2.0.100	
CoreRISC-V_AXI4		2.0.102	
Mi-V RV32		3.1.200	
Mi-V RV32		3.1.100	
Mi-V RV32		3.0.100	
Mi-V RV32IMAF_ L1_ AHB		2.1.100	
Mi-V RV32IMC		2.1.100	
No core selected			
Design Flow Design Hierarchy Stimulus Hierarch	y Cat	alog Components	Files

② Create Component ウインドウにて OK を押します。





③ CoreJTAGDebug IP の設定画面にて、デフォルトのまま OK を押します。

Configurator	-		×
CoreJTAGDebug Configurator Actel:DirectCore:COREJTAGDEBUG:4.0.100			
Configuration General Configuration Number of Debug Targets 1 UJTAG_BYPASS			
Debug_Target_0 Target 0 IR Code 0x55 Active-high target reset	Target 0 🗌]	
Debug_Target_1 Target 1 IR Code 0x56 Active-high target reset	Target 1 「	-	
Help	ок	Car	ncel

3-8. MIV_ESS

MIV_ESS を SmartDesign 上にドラッグ&ドロップします。
 注意: Mi-V ではなく 横棒(-)なしの MIV です。

miv	📃 🔍 🔻 📃 Simulation Mode	0 -
lame	△ Version	
Processors		
MIV_ESS	2.0.200	
MIV_ESS	2.0.100	
Mi-V RV32	3.1.200	
Mi-V RV32	3.1.100	
Mi-V RV32	3.0.100	
Mi-V RV32IMAF_ L1_ AHB	2.0.100	
Mi-V RV32IMAF_ L1_ AHB	2.1.100	
Mi-V RV32IMA_ L1_ AHB	2.0.100	
Mi-V RV32IMA_ L1_ AHB	2.3.100	
Mi-V RV32IMA_ L1_ AHB	2.1.100	
Mi-V RV32IMA_L1_AXI	2.1.100	
····· Mi-V RV32IMA_L1_AXI	2.0.100	
Mi-V RV32IMC	2.1.100	
o core selected		



② Create Component ウィンドウにて OK を押します。

Create Component	?	\times
Name:		
MIV_ESS_CO		
HelpOK	Car	ncel

③ GPIO 以外のチェックを外します

🕑 Configurator		D X
MIV_ESS ActelSystemBuilder:MIV_ESS2.0.200		
MIV_ESS_UI_default_configuration DGC1_PF_SPI_BOOT DGC2_PF_I2C_BOOT DGC3_PF_UPROM_BOOT DGC4_PF_BASIC_PERPHERALS	● General ● Bootstrap APB ● uDMA GPIO PLIC SPI ● Timer ● UART Family FPGA Family: PolarFire ● ● ● ● ● ● ● ● ● ● ● UART ● </td <td>]]</td>]]
Apply New preset		ن <i>…</i>
Help 👻	ок	Cancel

④ APB タブを開き、APB Mirrored I/F のチェックを外します。

Configurator							
MIV_ESS Actel:SystemBuilder:MIV_ESS:2.0.20	00						
MIV_ESS_UI_default_configurati DGC1_PF_SPI_BOOT	General General Bootstrap External APB Initiator APB Mirrored I/F:	APB	🚯 uDMA	GPIO	PLIC S	PI	1 Time
DGC3_PF_UPROM_BOOT DGC4_PF_BASIC_PERPHERALS	External APB Target	Slot 11 🗖	Slot 12 🗖	Slot 13 🗌	Slot 14 🗖	Slot	15 🗖
Apply New preset							
	•						▶
Help •					OK		Cancel



⑤ GPIO タブを開きます。

LED 用として、I/O bit 0~3の I/O Type を"Output"に設定し、OK を押します。 [◎] configurator

MIV_ESS		
Actel:SystemBuilder:MIV_ESS2.0.200		
MIV_ESS_UI_default_configuration	General Bootstrap APB UDMA GPIO PLIC SPI Timer UART Global Configuration	<u> </u>
DGC1_PF_SPI_BOOT DGC2_PF_I2C_BOOT DGC3_PF_uPROM_BOOT	APB Data Width: 32 Number of I/Os: 4	
- DGC4_PF_BASIC_PERPHERALS		
	I/O bit 1	
	I/O bit 2	
Apply New preset	Voltput on Reset: (0 · rived Connig (* 1/0 Type: Output · interrupt Type: Lesabled · I	
	Output on Reset: 0 Fixed Config: V I/O Type: Output Interrupt Type: Disabled I/O bit 4	
	Output on Reset: O Y Fixed Config: T 1/O Type: Input Y Interrupt Type: Disabled Y	ن
Help 🔻	OK	Cancel





3-9. ブロックの接続

SmartDesign 上のブロックを接続します。















【一括設定】 Ctrl キーや Shift キーの活用でピンの複数選択が可能です。
CORERESET_PF_C0_0
CUK EXT_RST_N BANK_X_VDDLSTATUS BANK_Y_VDDLSTATUS PLL_POWERDOWN_B- PLL_LOCK SS_BUSY INIT_DONE FF_US_RESTORE FFGA_POR_N CORERESET_PF_C0
【回路図の整形】
視認性が悪くなった場合 SmartDesign 上部の Reset Layout ボタンをクリックすると
回路図表示を整えることができます。
Image: Image with the provided and the pro
keset Layout



① PF_CCC_C0_0 ブロック

PF_CCC_C0_0	接続先
REF_CLK_0	ポートを出す(Promote to Top Level)
PLL_POWERDOWN_N_0	CORERESET_PF_C0_0: PLL_POWERDOWN_B
PLL_LOCK_0	CORERESET_PF_C0_0: PLL_LOCK
OUT0_FABCLK_0	MIV_RV32_C0_0 : CLK
	CORERESET_PF_C0_0 : CLK
	MIV_ESS_C0_0 : PCLK



② PFSOC_INIT_MONITOR_ C0_0 ブロック

PFSOC_INIT_MONITOR_	接続先
C0_0	
FABRIC_POR_N	CORERESET_PF_C0_0 : FPGA_POR_N
PCIE_INIT_DONE	Mask Unused
USRAM_INIT_DONE	Mask Unused
SRAM_INIT_DONE	Mask Unused
DEVICE_INIT_DONE	CORERESET_PF_C0_0: INIT_DONE
BANK_0_VDDI_STATUS	CORERESET_PF_C0_0 : BANK_x_VDDI_STATUS
	CORERESET_PF_C0_0 : BANK_y_VDDI_STATUS
XCVR_INIT_DONE	Mask Unused



USRAM_INIT_FROM_SNVM	Mask Unused
_DONE	
USRAM_INIT_FROM_UPRO	Mask Unused
M_DONE	
USRAM_INIT_FROM_SPI_D	Mask Unused
ONE	
SRAM_INIT_FROM_SNVM_	Mask Unused
DONE	
SRAM_INIT_FROM_UPROM	Mask Unused
_DONE	
SRAM_INIT_FROM_SPI_DO	Mask Unused
NE	
AUTOCALIB_DONE	Mask Unused





③ CORERESET_PF_C0_0 ブロック

PF_CCC_C0_0	接続先
EXT_RST_N	Tie High
SS_BUSY	Tie Low
FF_US_RESTORE	Tie Low
FABRIC_RESET_N	MIV_RV32_C0_0 : RESETN
	MIV_ESS_C0_0 : PRESETN



 ④ COREJTAGDEBUG_C0_0のJTAG_HEADER、DEBUG_TARGET_0、 MIV_RV32_C0_0のDEBUGの+ボタンをクリックし展開します。





⑤ COREJTAGDEBUG_C0_0 ブロック

COREJTAGDEBUG_C0_0	接続先
ТСК	ポートを出す(Promote to Top Level)
TDI	ポートを出す(Promote to Top Level)
TDO	ポートを出す(Promote to Top Level)
TMS	ポートを出す(Promote to Top Level)
TRSTB	ポートを出す(Promote to Top Level)
TGT_TCK_0	MIV_RV32_C0_0 : JTAG_TCK
TGT_TDI_0	MIV_RV32_C0_0 : JTAG_TDI
TGT_TDO_0	MIV_RV32_C0_0 : JTAG_TDO
TGT_TMS_0	MIV_RV32_C0_0 : JTAG_TMS
TGT_TRSTN_0	MIV_RV32_C0_0 : JTAG_TRSTN



⑥ MIV_ESS_C0_0 ブロック

MIV_ESS_C0_0	接続先
GPIO_IN[3:0]	Tie Low
APB_0_mINITIATOR	MIV_RV32_C0_0 : APB_INITIATOR
GPIO_OUT[3:0]	ポートを出す(Promote to Top Level)
GPIO_INT[3:0]	Mark Unused





⑦ MIV_RV32_C0_0 ブロック

MIV_RV32_C0_0	接続先
EXT_RESETN	Mark Unused
APB_M_TARGET	MIV_ESS_C0_0 : APB_0_TARGET



接続後:



⑧ Save します。

(Libero SoC プロジェクトを Save、もしくは回路図エディタタブを右クリックし Save)





⑨ Generate Component をクリックします。

Reports & X StartPage	₽× SD miv_top ₽×		
		× X X Q Q 🚱 Q 📰 A 🙋	4
Generate Component			

⑩ Design Hierarchy タブにて Build Hierarchy をクリックします。

Design Hierarchy	₽ ×
Top Module(root): miv_top	
Build Hierarchy	🔍 Show: Components 🖭 🗈 🔋 🗞
 Image: Book work Image: Book work work Image: Book work work work work work work work w	
Design Flow Design Hierarchy Stimu	Ilus Hierarchy Catalog Components Files



備考: Address map					
SmartDesign 上部の View Memory 1	Map 7	ドタンより、	Addre	ss map を確認	可能です。
SD mix ton 5 X Bornsto 5 X Start	Pogo A	x			
			57 Ve		
	=D ¹ =				
View Memory Map				<u>-F_CU_U</u>	
View Memory Map of miv_top				Box	
			5	Show Targets 🗌	
C 🖸 🖸 🔍 😣 🛦 🕕					
Initiator/Bus/Bridge/Target ∇ DRC	Offset A	ddress Range	High Ad	dress	
MIV_ESS_C0_0/MIV_APB3_0:APB3mmaster	0x0000	_0000 OB	0xFFFF_	FFFF	
Help					
長示は構成によって変わります。					
一例:					
/iew Memory Map				8)	
				Show Targets	
e • • 8 8 1					
Initiator/Bus/Bridge/Target	DRC	Offset Address	Range	High Address	
OreAHBLite_C0_0:AHB_MST_MMIO OreAHBLite_C0_0:AHBmmaster0					
COREAHBTOAPB3_C0_0:AHBslave		0x7000_0000	256MB	0x7FFF_FFFF	
CoreCoreCoreCoreCoreCoreCoreCoreCoreCore		0x7000_3000 0x7000_5000	4KB 4KB	0x7000_3FFF 0x7000_5FFF	
MIV_RV32IMA_L1_AHB_CO_0:AHB_MST_MEM		0.000_0000		0.000_000	
		0x8000_0008x0	64KB	0x8000_FFFF	
		0.4			


Address map のエラーが消えない場合:

現在 Libero SoC にて Issue があり、将来のバージョンにて修正される予定です。

引用「Important: In case any error is observed related to address space access issue, ensure to perform the following step: Navigate to Project > Project Settings > SmartDesign, and then Enable "Downgrade memory map generation DRC errors to warnings" as shown in the following figure. \Box

https://ww1.microchip.com/downloads/aemDocuments/documents/FPGA/ApplicationNo tes/ApplicationNotes/PolarFire_FPGA_Building_MIV_Subsystem_AN4997.pdf#page=21

🕙 Libero - C:¥miv_lab¥miv_top¥miv_top.prjx*					
Project File Edit View Design Tools Smart	Design Help				
New Project	Ctrl+N				
🚰 Open Project	Ctrl+O				
× Close					
💾 Save	Ctrl+Shift+S				
📙 Save C:¥miv_lab¥miv_top¥miv_top.prjx As	Ctrl+Shift+A				
📑 Archive Project					
🞯 Project Settings					
Tool Profiles					
Voult/Paparitarias Sattings					
Project cettings			_		~
Device settings Design flow Analysis operating conditio ⇒ Simulation options → DO file → Waveforms → Vsim commands ⊤ mescale ⇒ Simulation libraries → PolarFireSOC ← CORETAGDEBUG_LIB MIV_APB3_LIB → General Settings Global Include Paths SmartDesign	neration Design Rule Cl	heck(DRC) errors to warnings sign Rule Check(DRC) errors to warnings n Rule Check(DRC) errors to warnings		Discan	d
Help				Clos	se



3-10. Drive Constraints

① Design Flow タブから Manage Constraints を開き、

Timing タブにて Drive Constraints を実施します。

O Libero - C:¥miv_lab¥miv_top¥miv_top.prjx	- 🗆 X
Project File Edit View Design Tools Help	
] □ 🚰 🖬 🗠 □ ΙΟ 🖬 🗿 🗃	
Design Flow 🗗 🗙	Reports & X StartPage & Constraint Manager & Sommix_top & X
Top Module(root): miv_top 🖸 🖸 💕	I/O Attributes Timing Floor Planner Netlist Attributes
Active Synthesis Implementation: synthesis	New Import Link Edit V Check V Derive Constraints Constraint Coverage V
Tool	
🕀 🕨 Create Design	Synthesis Place and Route Timing Verification
Import MSS Greate SmartDesign	
Create HDL	
Create SmartDesign Testbench	
Create HDL Testbench	
Verify Pre-Synthesized Design	
Simulate	
P- Constraints	
- 🗟 Manage Constraints	
🕀 🕨 Implement Design	
🖓 Open Netlist Viewer	
Synthesize	
Verify Post-Synthesized Design	
• Generate Simulation File	
Simulate	
Configure Register Lock Bits	
Reace and Route	
Edit Doct Lavout Docign	
Design Flow Design Hierarchy Stimulus Hierarchy Catalog Components Files	

② Message ウィンドウにて Yes をクリックします。

Message	×
? To automatically associate the derived constraint SDC file to the 'Synthesis', 'Place and Route' and 'Timing Verification' tools click 'Yes' else click	'Noʻ.
Yes <u>N</u> o	

③ sdc ファイルが自動生成されたこと、Synthesis、Place and Route、

Timing Verification にチェックが入っていることを確認します。





3-11. 論理合成

Synthesize にて論理合成を実施します。



3-12. ピンアサイン

クロック、LED をピンアサインします。

ピン番号は、PolarFire SoC FPGA Discovery Kit User Guideの

3.5 Debug Circuitry

3.9 50 MHz Oscillator (DSC1001DL5-050.0000)

から確認可能です。

https://ww1.microchip.com/downloads/aemDocuments/documents/FPGA/ProductDocuments/UserGuides/PolarFire_SoC_FPGA_Discovery_Kit_User_Guide.pdf

① Constraint Manager の I/O Attributes タブより I/O Editor を開きます。





② 下記の通りアサインします。

信号名	ピン番号
GPIO_OUT[0]	T18
GPIO_OUT[1]	V17
GPIO_OUT[2]	U20
GPIO_OUT[3]	U21
REF_CLK_0	R18

③ Save して、I/O Editor を閉じます。

I/O Editor - C:/miv_lab/	miv_top (miv_top))				
File Edit View Logic	Tools Help					
	1 0	0 0		*	Ø.	S
Main Object Browser		Β×	Pir	n View	8	
	(∢ _			Port	Na
Ports	@ @	1	1	-	GPIO_	OU
🗄 🔮 💶 I/O Ports			2		GP	

Log ウィンドウにて

Error: PRPF_012: Instance 'PF_CCC_C0_0/PF_CCC_C0_0/pll_inst_0 (PLL_IP)' must be placed and locked at a legal location before running Place and Route.

が表示された場合無視します。(後ほど配置配線にて自動で PLL をアサインします。)

.og																									
Message	s 🔇	Errors	🔥 Warnings	•	Info																				
OInfo: 1	New :	IO con	straints	were	writte	n to f	file (:/miv	lab/m	iv t	op/cor	nstra	aint/i	io/p	re/tmp	extr	a cst	52846	3.pdc						
DError:	PRPI	F_012:	Instance	'PF	CCC_C0	0/PF	CCC 0	0_0/p	l inst	: 0	(PLL_I	[P)'	must	be	placed	and	locked	at a	legal	location	before	running	Place #	and Re	oute.
DError:	PRPI	F 012:	Instance	'PF	CCC CO	0/PF	CCC 0	:0 ⁻ 0/p:	l inst	: 0	(PLL I	[P)'	must	be	placed	and	locked	at a	legal	location	before	running	Place /	and Re	oute.
@Error:	PRPI	F 012:	Instance	'PF	CCC CO	0/PF	CCC 0	c0 0/p:	l inst	: 0	(PLL I	(P)	must	be	placed	and	locked	at a	legal	location	before	running	Place /	and Re	oute.
DError:	PRPI	012:	Instance	'PF	CCC CO	0/PF	CCC 0	0 0/p	linst	: 0	(PLL I	(P)	must	be	placed	and	locked	at a	legal	location	before	running	Place /	and R	oute.
OInfo: 1	Desi	gn Rul	es Check	comp.	leted s	uccess	sfully		-	-	· -														
GInfo: 1	New 3	IO con	straints	were	writte	n to f	file (:/miv	lab/mi	iv_t	op/cor	nstra	aint/i	io/u	ser.pdd	0									
										-															
Log Mess	age																								



④ (もしチェックが入っていない場合)

Place and Route ヘチェック入れて Save します。

Reports & X StartPage & X	Constraint Manager* & X SD miv_top & X	Ŧ
I/O Attributes* Timing Floor Planner	Netlist Attributes	
New 🔽 Import 🛛 Link	Edit 👻 View Check Help	Save Discard
constraint¥io¥user.pdc	Place and Route	

Warning ウィンドウが出た場合は OK をクリックします。

larning ×
Saving the changes in the Constraint Manager tool may invalidate your design flow. You may have to rerun your design flow. Do you want to save the changes? Don't show again.
OK Cancel

3-13. 配置配線

Place and Route をクリックし、配置配線します。

Design Flow				đΧ
Top Module(root): miv_top	-	0	и 🛛	\$
Active Synthesis Implementation: synthesis				
Tool				•
Create HDL				
Create HDL Testbench				
🖃 🕨 🕨 Verify Pre-Synthesized Design				
Simulate				
Constraints				
Manage Constraints				
🗊 🕨 Implement Design				
🖓 Open Netlist Viewer				
V Synthesize				
🖃 🕨 🕨 Verify Post-Synthesized Design				
Generate Simulation File				
Simulate				
Configure Register Lock Bits				
Place and Route				
Edit Post Layout Design				
Verify Post Layout Implementation				
Generate Back Annotated Files				_
	_	_		<u> </u>
Design Flow Design Hierarchy Stimulus Hierarchy Cata	log Co	mponent	s Fil	es



3-14. 書き込み

Discovery Kit を PC へ接続し、Run PROGRAM Action をダブルクリック、 作成したデザインを書き込みます。





4. ソフトウェア

4-1. SoftConsole 起動

SoftConsole を起動します。



- 4-2. workspace の作成
- SoftConsole のプロジェクトを管理する Workspace として任意のフォルダを指定、 Launch をクリックします。

SC SoftConsole vRISC-V- Launcher	×
Select a directory as workspace	
SoftConsole v -RISC-V- uses the workspace directory to store its preferences and development artifacts.	
Workspace: C:¥miv_lab¥miv_top¥SoftConsole <u>B</u> rowse	
Use this as the default and do not ask again	
Becent Workspaces	
· Tecent workshaces	
<u>L</u> aunch Cancel	



 New Directory ウィンドウにて、新しく workspace を作成するかどうか聞かれるため OK をクリックします。

sc New	v Directory	×
8	Do you want to create a new workspace?	
	OK Cancel	

③ Welcome ページを閉じます。



Welcome ページを閉じた後:





4-2. Driver の入手

① GitHub にて "Mi V Soft RISC V" ページを開きます。 https://github.com/Mi-V-Soft-RISC-V



② ページを下にスクロールします。

Mi V CPU やペリフェラルの Driver を入手するため Platform のリンクを開きます。

Bare Metal Embedded Software

- Platform: Hardware Abstraction Layer (HAL) and peripheral drivers for Mi-V Soft CPUs
- <u>Mi-V RV32 Bare Metal Examples</u>: drivers and example projects for Mi-V Soft RISC-V CPUs and their associated peripherals

備考: Mi-V RV32 Bare Metal Examples

今回は 0 から C ソースを作成しますが、Mi V RV32 Bare Metal Examples を ダウンロードするとペリフェラルに応じたサンプルデザインが含まれています。 必要に応じて適宜ご参考ください。

Bare Metal Embedded Software

- Platform: Hardware Abstraction Layer (HAL) and peripheral drivers for Mi-V Soft CPUs
- Mi-V RV32 Bare Metal Examples: drivers and example projects for Mi-V Soft RISC-V CPUs and their associated peripherals



③ 最新版をダウンロードします。

Product ~ So		ices opensou							<u> </u>	
Mi-V-Soft-RISC	-V / platform(Public			🗘 Notif	ications	양 Fork	1	☆ Star	3
> Code 💿 Issues	1 រឺរិ Pull reque	sts 🕑 Actions [Projects	() Security	🗠 Insight	ts				
វិ main 👻 វិ	\Diamond	Q Go to file		<> Code	Abo	out				
👸 aakash-mchp Mov	e MTVEC_BASE_ADD	R_MA 🚥 4cf9547	· 6 months ago	🕙 133 Commite	No	<i>description</i> Readme	n, websii	te, or top	oics prov	ided.
drivers/fpga_ip	up	date for New ECC Inter	rupts	8 months ag	হা হ	View licens	se			
🖿 hal	Me	rged hal source code v	version 1.0	3 years ag	\$	Code of co	onduct			
miv_rv32_hal	Mc	ove MTVEC_BASE_ADDF	R_MASK o	6 months ag	· · ·	Activity Custom pr	operties			
LICENSE.md	ad	d LICENSE.md file to the	e platfor	2 years ag	, ☆	3 stars				
README.md	rea	dme : Use correct drive	er folder n	4 years ag	•	10 watchin	g			
	de efferenduet atta				Repo	1 fork ort reposito	ory			
code										
leases / 2024.09 2024.09 Singh-Raghvendr Mi-V Soft Pr CoreTSE • Optional s • Incorpora • CoreQSPI	(Latest) a released this Oct OCESSOT Pla separate interrup te support for EC	25, 2024 🔊 2024 Itform t handlers for TX ar C error handling.	.09 -0- 4c	:f9547	+ 7	releases			Ca	ompare
leases / 2024.09 2024.09 Singh-Raghvendr Mi-V Soft Pr CoreTSE • Optional s • Incorpora • CoreQSPI • First Relea • MIV_RV32_HA • Fixed an is	Latest a released this Oct cocessor pla separate interrup te support for EC use L ssue where the M	25, 2024 🔊 2024 Itform t handlers for TX ar C error handling. ITVEC_BASE_ADDR	.09 - 40 nd RX.	:f9547 cro was unde	+ 7	n MIV_R\	/32_EX1	ſ_TIMEF	Cr cr t was d	pmpare
leases / 2024.09 2024.09 Singh-Raghvendr Mi-V Soft Pr CoreTSE • Optional s • Incorpora • CoreQSPI • First Relea • MIV_RV32_HA • Fixed an is Driver	Latest a released this Oct cocessor pla separate interrup te support for EC use L ssue where the M Revision	25, 2024 🔊 2024 Itform t handlers for TX ar C error handling. ITVEC_BASE_ADDR	.09 -0- 4c nd RX.	:f9547 cro was unde	+ 7	n MIV_R\	/32_EX1	ſ_TIMEF	co R was d	efined
eases / 2024.09 2024.09 Singh-Raghvendr Mi-V Soft Pr CoreTSE Optional s Optional s First Relea MIV_RV32_HA Fixed an is Driver CoreTSE CoreQSPI	Latest a released this Oct a released this Oct cocessor pla separate interrup te support for EC ase L ssue where the M Revision 2.6.001 2.1.103	25, 2024	.09 -0- 4c	:f9547 cro was unde	+ 7	n MIV_R\	/32_EX1	ſ_TIMEF	Cc R was d	ompare
leases / 2024.09 2024.09 Singh-Raghvendr Mi-V Soft Pr • CoreTSE • Optional s • Incorpora • CoreQSPI • First Relea • MIV_RV32_HA • Fixed an is Driver CoreTSE CoreQSPI × Assets 2	L separate interrup te support for EC sesue where the M Revision 2.6.001 2.1.103	25, 2024 S 2024 Itform t handlers for TX ar C error handling. ITVEC_BASE_ADDR	.09 -0- 4c	:f9547 cro was unde	ined whe	n MIV_R\	/32_EX1	r_TIMEF	Ca R was d	ompare
leases / 2024.09 2024.09 Singh-Raghvendr Mi-V Soft Pr • CoreTSE • Optional s • Incorpora • CoreQSPI • First Relea • MIV_RV32_HA • Fixed an is Driver CoreTSE CoreQSPI ▼ Assets 2 Source code (zig	Latest a released this Oct OCESSOR pla separate interrup te support for EC use L ssue where the M 2.6.001 2.1.103	25, 2024 2024 Itform t handlers for TX ar C error handling. ITVEC_BASE_ADDR	.09 -0- 4c	:f9547 cro was unde	ined whe	n MIV_R\	/32_EX1	[_TIMEF	C ca R was d	efinec 24, 20



④ ダウンロードした zip ファイルを任意の場所に解凍します。



以前は Firmware Catalog から Driver を入手していましたが 現在は GitHub から入手頂くよう変更になりました。	備考: Firmware Cata	logにこ	ついて						
現在は GitHub から入手頂くよう変更になりました。 第firmware Catalog × Firmware Catalog × Firmware Catalog × Fire View Tools Help Vew (52/175): Search by all fields (53/53): Vew (52/175): Search by all fields (53/175): Vew (52/175): Vew (52/175): Vew (52/175): Vew (52/175): Vew (52/175): Vew (5	以前は Firmware Car	talog カ	کر Drive	r を入手	ミレている	きした	こが		
見在は GitHub から入手頂くよう変更になりました。 ■ firmware Catalog		uiog ~				201	- /0		
Pirmware Catalog	見在け GitHub から	、入手頂	くよう変更	更にたり	キレた	•			
Firmware Catalog ✓ Yew (50:175): Search by all fields (50:25): ✓ All ✓ Vew (50:175): Search by all fields (50:25): ✓ All ✓ Vexit ✓ Vexit		×/ 1 17			\$ 0,00	.0			
Core Mindor Cettalog View (53/175): Search by all fields (53/53): i View (53/175): Search by all fields (53/53): View (53/175): i View (53/175): Status CoreMMC Driver 2.0.100 0.62 CoreFWM Driver 9.9.100 0.40 CoreSyStervices PP Driver 9.9.100 0.39 CoreUARTapb Driver 9.9.100 0.39 CoretWA Dog Driver 9.9.100 0.39 CoretWA Dog Driver 9.9.100 0.39 CoretWA Dog Driver 9.9.100 0.40 CoretWA Dog Driver 9.9.100 0.40 CoretWA Dog Driver 9.9.100 0.40 CoretWA Dog Driver	Sirmware Catalog			_					
View (52/175): View (52/175):	ile View Tools Help								
Vew (62/1/5): Starth by all heids (52/55): Vew (62/1/5): Image: Starth by all heids (52/55): Name Version Starth by all heids (52/55): CoreMXC Driver 30.101 1.54 CoreMXC Driver 20.100 0.62 CoreSPL Driver 9.9.100 0.39 CoreTSE Driver 9.9.100 0.40 CoreTVactAR Dip Driver 9.9.100 0.39 CoreUAtARdop Driver 9.9.100 0.40 CoreWatArdong Driver 9.9.100 0.40 CoreVactArdop Driver 9.9.100 0.40 CoreVactArdop Driver 2.1.101 3.16 Hardware Abstraction Layer (HAL) 2.3.102 0.22 View Vew Heide In the "platform" repository of the MI-V Soft RISC-V CoreTimer bare metal driver is included in the "platform" repository of the MI-V Soft RISC-V RISC-V/MI-VS	(50 (175))				(50.(50))	_			
✓ display only the latest version of a core Name Version Size (MB) CoreMMC Driver 3.0.101 1.54 CoreMMC Driver 2.0.100 0.62 CoreSPL Driver 9.9.100 0.39 CoreSPL Driver 9.9.100 0.40 CoreSPL Driver 9.9.100 0.40 CoreSPL Driver 9.9.100 0.40 CoreSPL Driver 9.9.100 0.40 CoreWARtapb Driver 9.9.100 0.40 CoreWARtapb Driver 9.9.100 0.39 CoretImer Driver 9.9.100 0.40 CoreWARtapb Driver 9.9.100 0.22 Variational CoreSPL Driver 9.9.100 0.40 CoreTimer bare metal driver and its examples are now being delivered via GitHub CoreTimer bare metal driver is included in the "	View (<u>53/1 /5</u>):		Searc	<u>h</u> by all fields i	. <u>53753</u>):				
✓ display only the latest version of a core Name / Version Size (MB) Status CoreMMC Driver 3.0.101 1.54 CoreMMD Driver 2.0.100 0.62 CoreSUD Driver 2.1.100 1.27 CoreSystervices, PF Driver 9.9.100 0.40 CoreTimer Driver 9.9.100 0.40 CoreUARTapb Driver 9.9.100 0.39 CoreUARTapb Driver 9.9.100 0.39 CoreUARTapb Driver 9.9.100 0.40 CoreWatchdog Driver 2.2.100 0.50 CoreUARTapb Driver 9.9.100 0.29 CoreTimer Driver 9.9.100 0.20 CoreWatchdog Driver 2.2.100 0.50 CoretVartAdog Driver 2.3.102 0.22 Vit Description: This firmware driver and its examples are now being delivered via GitHub. Core Timer bare metal driver is included in the "platform" repository of the M=V Soft RISC-V V GitHub organisation https:// github.com/M=V=Soft-RISC-V/Diatform. Core Timer bare metal example is included in the "miv-rv32-bare-metal-examples" RISO-V Miv-V-Soft-RISC-V GitHub organisation https:// github.com/ platform". <t< td=""><td>🕺 🏹 🕺 Vault 🔮 We</td><td>b repositories</td><td>ri</td><td></td><td> •</td><td></td><td></td><td></td><td></td></t<>	🕺 🏹 🕺 Vault 🔮 We	b repositories	ri		•				
Name / Version Size (MB) Status CoreMMC Driver 30.101 1.54 CoreMMC Driver 20.100 0.62 CorePWM Driver 99.100 0.39 CoreSDL Driver 21.100 1.27 CoreSPI Driver 99.100 0.40 CoreSPE Driver 99.100 0.39 CoreUner Driver 99.100 0.39 CoreUner Driver 99.100 0.39 CoreUner Driver 99.100 0.39 CoreUner Driver 99.100 0.40 CoreWatchdog Driver 22.100 0.50 Cortex-MT CMSIS Hardware Abs 21.101 3.16 Hardware Abstraction Layer (HAL) 23.102 0.22 Virtual CoreTimer bare metal driver is included in the "platform" repository of the MI-V Soft RISC-V/platform. CoreTimer bare metal arkample is included in the "miv-rk92-bare-metal-examples" repository of the MI-V Soft RISC-V (Mitub organisation https://github.com/MI-V-Soft-RISC-V/platform. CoreTimer bare metal example is included in the "miv-rk92-bare-metal-examples" Repository of the MI-V Soft RISC-V (Mitub organisation https://github.com/Mi-V-Soft-RISC-V/platform. Virtual CoreTimer bare metal examples. Pola	✓ display only the latest version of	a core	,						
CoreMMC Driver 3.0.101 1.54 CoreMMC Driver 2.0.100 0.62 CoreSUC Driver 9.9.100 0.39 CoreSDI Driver 9.9.100 0.40 CoreSSPE Driver 9.9.100 0.40 CoreTSE Driver 9.9.100 0.39 CoreUMARTapb Driver 9.9.100 0.39 CoreUMARTapb Driver 9.9.100 0.40 CoreWatchdog Driver 2.2.100 0.50 Cortex-M1 CMSIS Hardware Abs 2.1.101 3.16 Hardware Abstraction Layer (HAL) 2.3.102 0.22 Description: This firmware driver and its examples are now being delivered via GitHub. CoreTimer bare metal driver is included in the "platform" repository of the MI-V Soft RISO- V GitHub organisation https://github.com/MI-V-Soft-RISO- V GitHub organisation https://github.com/MI-V-Soft-RISO- V GitHub organisation https://github.com/MI-V-Soft-RISO- V GitHub organisation https://github.com/MI-V-Soft-RISO- CoreTimer bare metal examples. PolarFire SoC IP core bare metal drivers are included in the "platform" repository of the PolarFire SoC GitHub organisation https://github.com//platform.	Name 🛆	Version	Size (MB)	Status					
CoreMMC Driver 2.0.100 0.62 CoreSPWM Driver 9.9.100 0.39 CoreSDLC Driver 2.1.100 1.27 CoreSP Driver 9.9.100 0.40 CoreSysServices_PF Driver 9.9.100 0.39 CoreLARTapb Driver 9.9.100 0.40 CoreUARTapb Driver 2.2.100 0.50 CoreUARTapb Driver 2.2.100 0.50 Coretwatchdog Driver 2.2.100 Ocae Timer bare metal driver is included in the "platform" repository of the MI-V Soft RISC-V/platform. Core Timer bare metal example is included in the "miv-rv32-bare-metal-examples" repository of the MI-V Soft RISC-V GitHub organisation https://github.com/MI-V-Soft- RISC-V/miv-rv32-bare-metal-examples. PolarFire SoC IP core bare metal drivers are included in the "platform" repository of the PolarFire SoC GitHub organisation https://github.com/platfire-soc/platform.	CoreMMC Driver	3.0.101	1.54						
CorePWM Driver 9.9.100 0.39 CoreSDLC Driver 2.1.100 1.27 CoreSPI Driver 9.9.100 0.40 CoreSystevices_PF Driver 9.9.100 0.39 CoreTimer Driver 9.9.100 0.39 CoreUARTapb Driver 9.9.100 0.39 CoreUARTapb Driver 2.2.100 0.50 Cortex-M1 CMSIS Hardware Abs 21.101 3.16 Hardware Abstraction Layer (HAL) 2.3.102 0.22 Description: This firmware driver and its examples are now being delivered via GitHub. Core Timer bare metal driver is included in the "platform" repository of the MI–V Soft RISC- V GitHub organisation https://github.com/MI–V–Soft- RISC-V/miv–rv32-bare-metal-examples. PolarFine SoC IP ore bare metal drivers are included in the "platform" repository of the PolarFine SoC GitHub organisation https://github.com/platfire-soc/platform.	CoreMMC Driver	2.0.100	0.62			1			
CoreSDLC Driver 2.1.100 1.27 CoreSPI Driver 9.9.100 0.40 CoreSysServices_PF Driver 9.9.100 0.39 CoreUARTapb Driver 9.9.100 0.39 CoreUARTapb Driver 9.9.100 0.40 CoreWatchdog Driver 2.2.100 0.50 Cortex-M1 CMSIS Hardware Abs 2.1.101 3.16 Hardware Abstraction Layer (HAL) 2.3.102 0.22 V GItHub organisation https://github.com/MI-V-Soft RISO- V GitHub organisation https://github.com/MI-V-Soft-RISO- CoreTimer bare metal driver is included in the "platform" repository of the Mi-V Soft RISO- CoreTimer bare metal enversare included in the "miv-rv32-bare-metal-examples" repository of the Mi-V Soft RISO-V GitHub organisation https://github.com/MI-V-Soft- RISO-V/miv-rv32-bare-metal-examples. PolarFine SoC IP core bare metal drivers are included in the "platform" repository of the PolarFine SoC IP core bare metal drivers are included in the "platform" repository of the PolarFine SoC IP core bare metal drivers are included in the "platform" repository of the PolarFine SoC IP core bare metal drivers are included in the "platform" repository of the PolarFine SoC IP core bare metal drivers are included in the "platform" repository of the PolarFine SoC GitHub organisation https://github.com/polarFine-soc/platform.	CorePWM Driver	9.9.100	0.39						
CoreSPI Driver 9.9.100 0.40 CoreSysServices_PF Driver 9.9.100 0.39 CoreTimer Driver 9.9.100 0.39 CoreUARTapb Driver 9.9.100 0.40 CoreWatchdog Driver 2.2.100 0.50 Cortex-M1 CMSIS Hardware Abs 21.101 3.16 Hardware Abstraction Layer (HAL) 2.3.102 0.22 "" Description: This firmware driver and its examples are now being delivered via GitHub. CoreTimer bare metal driver is included in the "platform" repository of the Mi–V Soft RISO- V GitHub organisation https://github.com/MI–V–Soft-RISO- V/miv–N32-bare-metal-examples. PolarFine SoC IP core bare metal drivers are included in the "platform" repository of the PolarFine SoC GitHub organisation https://github.com/polarfire-soc/platform.	CoreSDLC Driver	2.1.100	1.27						
CoreSySServices PF Driver 9.9.100 0.40 CoreTSE Driver 9.9.100 0.39 CoreUARTapb Driver 9.9.100 0.40 CoreWatchdog Driver 2.2.100 0.50 Coretwatchdog Driver 2.2.100 Coretwatchdog D	CoreSPI Driver	9.9.100	0.40			-			
CoreTSE Driver 9.9.100 0.39 CoreTSE Driver 9.9.100 0.39 CoreUARTapb Driver 9.9.100 0.40 CoreWatchdog Driver 2.2.100 0.50 Cortex-M1 CMSIS Hardware Abs 2.1.101 3.16 Hardware Abstraction Layer (HAL) 2.3.102 0.22 The core Timer bare metal driver is included in the "platform" repository of the MI–V Soft RISC- V GitHub organisation https://github.com/MI–V–Soft-RISC-V/platform. Core Timer bare metal driver is included in the "miv-rv32-bare-metal-examples" repository of the MI–V Soft RISC-V GitHub organisation https://github.com/MI–V–Soft- RISC-V/miv-rv32-bare-metal-examples. PolarFire SoC IP core bare metal drivers are included in the "platform" repository of the PolarFire SoC GitHub organisation https://github.com/polarfire-soc/platform.	CoreSysServices_PF Driver	9.9.100	0.40						
CoreTimer Driver 9.9.100 0.39 CoreUARTapb Driver 9.9.100 0.40 CoreWatchdog Driver 2.2.100 0.50 Cortex-M1 CMSIS Hardware Abs 2.1.101 3.16 Hardware Abstraction Layer (HAL) 2.3.102 0.22 Image: CoreTimer bare metal driver and its examples are now being delivered via GitHub. Image: CoreTimer bare metal driver is included in the "platform" repository of the Mi-V Soft RISO-V/gitHub.com/MI-V-Soft-RISO-V/platform. CoreTimer bare metal example is included in the "miv-rv32-bare-metal-examples" repository of the Mi-V Soft RISO-V/gitHub.com/MI-V-Soft-RISO-V/gitHub.com/MI-V-Soft-RISO-V/platform. CoreTimer bare metal example is included in the "miv-rv32-bare-metal-examples" repository of the Mi-V Soft RISO-V/miv-rv32-bare-metal-examples" PolarFine SoC IP core bare metal drivers are included in the "platform" repository of the PolarFine SoC GitHub organisation https://github.com/polarFine-soc/platform. Image: Download Image: Download	CoreTSE Driver	9.9.100	0.39						
CoreUARTapb Driver 9.9.100 0.40 CoreWatchdog Driver 2.2.100 0.50 Cortex-M1 CMSIS Hardware Abs 21.101 3.16 Hardware Abstraction Layer (HAL) 2.3.102 0.22 W Core Timer bare metal driver and its examples are now being delivered via GitHub. Core Timer bare metal driver is included in the "platform" repository of the MI–V Soft RISC– V GitHub organisation https://github.com/MI–V–Soft-RISC–V/platform. Core Timer bare metal example is included in the "miv-rv32-bare-metal-examples" repository of the MI–V Soft RISC–V GitHub organisation https://github.com/MI–V–Soft- RISC–V/miv-rv32-bare-metal-examples. PolarFine SoC IP core bare metal drivers are included in the "platform" repository of the PolarFine SoC GitHub organisation https://github.com/polarFire-soc/platform.	CoreTimer Driver	9.9.100	0.39						
CoreWatchdog Driver 2.2.100 0.50 Cortex-M1 CMSIS Hardware Abs 2.1.101 3.16 Hardware Abstraction Layer (HAL) 2.3.102 0.22	CoreUARTapb Driver	9.9.100	0.40						
Cortex-M1 CMSIS Hardware Abs 2.1.101 3.16 Hardware Abstraction Layer (HAL) 2.3.102 0.22	CoreWatchdog Driver	2.2.100	0.50						
Hardware Abstraction Layer (HAL) 2.3.102 0.22	Cortex-M1 CMSIS Hardware Abs	2.1.101	3.16						
Description: This firmware driver and its examples are now being delivered via GitHub. Core Timer bare metal driver is included in the "platform" repository of the MI–V Soft RISC- V GitHub organisation https://github.com/MI–V-Soft-RISC-V/platform. Core Timer bare metal example is included in the "miv-rv32-bare-metal-examples" repository of the MI–V Soft RISC-V GitHub organisation https://github.com/MI–V-Soft- RISC-V/miv-rv32-bare-metal-examples. PolarFire SoC IP core bare metal drivers are included in the "platform" repository of the PolarFire SoC GitHub organisation https://github.com/polarfire-soc/platform.	Hardware Abstraction Layer (HAL)	2.3.102	0.22		•	1			
Description: This firmware driver and its examples are now being delivered via GitHub. CoreTimer bare metal driver is included in the "platform" repository of the Mi–V Soft RISC– V GitHub organisation https://github.com/Mi–V–Soft–RISC–V/Diafform. CoreTimer bare metal example is included in the "miv–rx92–bare-metal-examples" repository of the Mi–V Soft RISC–V GitHub organisation https://github.com/Mi–V–Soft– RISC–V/miv–rx92–bare-metal-examples. PolarFire SoC IP core bare metal drivers are included in the "platform" repository of the PolarFire SoC GitHub organisation https://github.com/polarfire-soc/platform.	4	0.0.400	0.04			-			
Description: I his firmware driver and its examples are now being delivered via GitHub. Core Timer bare metal driver is included in the "platform" repository of the Mi–V Soft RISC- V GitHub organisation https://github.com/Mi–V–Soft–RISC-V/platform. Core Timer bare metal example is included in the "miv–rv32–bare-metal-examples" repository of the Mi–V Soft RISC-V GitHub organisation https://github.com/Mi–V–Soft- RISC-V/miv–rv32–bare-metal-examples. PolarFire SoC IP core bare metal drivers are included in the "platform" repository of the PolarFire SoC GitHub organisation https://github.com/polarfire-soc/platform. @ Download @ Generate						Т			
Core Timer bare metal driver is included in the "platform" repository of the Mi–V Soft RISC- V GitHub organisation https://github.com/Mi–V–Soft–RISC-V/platform. Core Timer bare metal example is included in the "miv–rv32-bare-metal-examples" repository of the Mi–V Soft RISC-V GitHub organisation https://github.com/Mi–V–Soft- RISC-V/miv–rv32-bare-metal-examples. PolarFire SoC IP core bare metal drivers are included in the "platform" repository of the PolarFire SoC GitHub organisation https://github.com/polarFire-soc/platform.	Description: This firmware driver a	and its example	es are now being de	elivered via Git	Hub.	-			
CoreTimer bare metal example is included in the "miv-rv32-bare-metal-examples" repository of the MI-V Soft RISC-V GitHub organisation https://github.com/MI-V-Soft- RISC-V/miv-rv32-bare-metal-examples. PolarFire SoC IP core bare metal drivers are included in the "platform" repository of the PolarFire SoC GitHub organisation https://github.com/polarfire-soc/platform.	CoreTimer bare metal driver is inclu V GitHub organisation https://githuk	ded in the ″pl b.com/Mi=V=S	atform″repository Soft-RISC-V/platfo	of the Mi−VS rm.	oft RISC-				
repository of the Mi-V Soft RISC-V GitHub organisation https://github.com/Mi-V-Soft- RISC-V/miv-rv32-bare-metal-examples. PolarFire SoC IP core bare metal drivers are included in the "platform" repository of the PolarFire SoC GitHub organisation https://github.com/polarfire-soc/platform.	CoreTimer bare metal example is in	cluded in the '	″ miv=rv32=bare=me	etal-examples"		1			
PolarFire SoC IP core bare metal drivers are included in the "platform" repository of the PolarFire SoC GitHub organisation https://github.com/polarfire-soc/platform.	repository of the Mi-V Soft RISC-V RISC-V/miv-rv32-bare-metal-exam	'GitHuborgar nples.	isation https://gith	ub.com/Mi-V-	Soft-				
Download Cenerate	PolarFire SoC IP core bare metal dr PolarFire SoC GitHub organisation h	ivers are inclu https://github.	uded in the ″platfor com/polarfire-soc/	m″repository ′platform.	of the	1			
			6	Download	🛗 Generate	1			
			_			-			
					11	h			



4-3. 新規プロジェクト作成

① File > New > Project... $p \neq 0$

sc SoftConsole - SoftConsole v2022.2-RISC-V-747

File	Edit	Source	Refactor	Navigate	Search	Project	Gi	t Run	Window	Help)	
	New				Alt+Sh	nift+N≯	C ++	C++ P	roject			
	Open	File					¢	C Proje	ect			
0	Open	Projects	from File S	ystem			C	C/C++	Project			
					-		C++	Makef	ile Proiect	with E	xistina Code	
	Close	Editor			C	trl+W	D)	Project	t			
	Close	All Edito	rs		Ctrl+Sh	ift+W						
	Save				(Ctrl+S	C b	Source	e File er File		New Project	

② C/C++を展開し、 C Project を選択して Next をクリックします。

SC New Project			ı x
Select a wizard			-
Create a new C project			
<u>W</u> izards:			
type filter text			
 > Seneral > C/C++ C/C++ Project C++ Project C Project Makefile Project with Existing Code > Robot Framework 			
? < <u>B</u> ack <u>N</u> ext > Ei	nish	Cá	ancel



③ 下記の通り設定し Next をクリックします。

Project name : MiV_SW

Project type : Empty Project を選択

Toolchains: RISC V Cross GCC を選択

sc C Project				×
C Project				-
Create C project of selected type				_
Project name: MiV_SW				
✓ Use <u>d</u> efault location	VA 43 4 63 44		5	
Location: C:¥miv_lab¥miv_top¥SoftConsole	**MIV_SW		B <u>r</u> owse	
Project type:	Toolchains:			
 Executable Empty Project Hello World RISC-V C Project Shared Library Static Library Makefile project Show project types and toolchains only in 	RISC-V Cross GCC	n the	platform	
? < <u>B</u> ack <u>N</u> ext >	<u>F</u> inish		Cancel	

④ デフォルトのまま Next ボタンをクリックします。

sc C Project			×
Select Configurations Select platforms and configurations you wish to deploy on			Ď
Project type: Executable Toolchains: RISC-V Cross GCC Configurations:			
 ✓ Stopped Debug ✓ Stopped Release 		Select Deselec	all t all
	Adv	anced se	ettings
Use "Advanced settings" button to edit project's properties. Additional configurations can be added after project creation. Use "Manage configurations" buttons either on toolbar or on pr	roperty	pages.	
? < <u>B</u> ack <u>N</u> ext > Einish		Car	ncel



⑤ デフォルトのまま Finish ボタンをクリックします。

SC C Project			
GNU RISC-V C Select the toolc	r oss Toolchain nain and configure path		Ď
Toolchain name:	RISC-V GCC/Newlib (riscv64-unknown-elf-	gcc)	
Toolchain path:	<pre>\${eclipse_home}//riscv-unknown-elf-gcc/</pre>	bin	Browse.
?	< <u>Back</u> Next > <u>Finis</u>	sh	Cancel

⑥ プロジェクトが作成されたことを確認します。

Sc SoftConsole - SoftConsole v2022.2-RISC-V-747 File Edit Source Refactor Navigate Search External Tools V Mi-V-Renoc Project Explorer × EST & D MiV_SW



4-4. Driver のインポート

① SoftConsole にて MiV_SW を右クリックし Import... をクリックします。

陷 Project Exp	lore	r × 🕒 🔄 🍞 🖇 🗖 🗖	
🔸 😂 MiV_SM	/		
		New Go Into	>
		Open in New Window	
		Show In	Alt+Shift+W >
		Show in Local Terminal	>
		Сору	Ctrl+C
	Ē	Paste	Ctrl+V
	×	Delete	Delete
		Source	>
		Move	
		Rename	F2
	è	Import Import	
	4	Export	
	٨	Robot Framework	>
		Puild Project	

② File System を選択し Next をクリックします。





 Browse...ボタンから、GitHubからダウンロードし解凍した platform-xxxx フォルダを 選択します。

sc Import			_	
File system Source must not	be empty.			
From directory:			~	B <u>r</u> owse
Filter <u>T</u> ypes	<u>S</u> elect All	Deselect All		
Into fo <u>l</u> der: MiV_	SW			Bro <u>w</u> se
Options Qverwrite exis Create top-lev Advanced >>	sting resources witho vel folder	out warning		
?	< <u>B</u> ack	<u>N</u> ext >	<u>F</u> inish	Cancel



④ 下記ヘチェックを入れ、 Finish をクリックします。 drivers/ fpga_ip/CoreGPIO

hal

miv_rv32_hal

sc Import		-		×
File system				
Import resources from the local file system.				
From directory: C:¥platform-		~	B <u>r</u> o	wse
✓ ■ ▷ platform-	LICENSE.n	nd		
✓ ■ 🗁 drivers	README.r	nd		
✓ ■ ► Tpga_Ip				
Core TUGBaseKK_PHY				
> CoreUARTapb				
\rightarrow $\square \supseteq$ coreor at tapp				
\rightarrow $\square \supseteq min_2 \square \square$				
> C C miv_pine				
> C C miv_aand				
$\rightarrow \checkmark \rightleftharpoons \Leftrightarrow min$ rv32 hal				
Filter Types Select All Deselect All				
Into folder: MiV_SW			Bro	<u>w</u> se
Options				
Overwrite existing resources without warping				
<u>A</u> dvanced >>				
? < <u>B</u> ack	<u>N</u> ext >	<u>F</u> inish	Ca	ncel

インポート後:





4-5. プロパティ設定

① プロジェクト名を右クリックし、Properties をクリックします。

陷 Project Exp	lorer	× 🗉 🔄 🏹 🖇 🗖 🗖	
 ✓ [™] MiV_SW > [™] Inclu > [™] drive 		New Go Into	>
> 🗁 hal > 🗁 miv_ı		Open in New Window Show In Show in Local Terminal	Alt+Shift+W > >
		Copy Paste Delete Source Move	Ctrl+C Ctrl+V Delete
		Rename	F2
	<u>s</u>	Export	
	8	Robot Framework	>
	Ł	Build Project Clean Project Refresh Close Project Close Unrelated Project	F5
		Build Configurations	>
		Build Targets Index	>
<		Profiling Tools	>
🎋 Debug 🗵	•	Run As	>
🗆 💥 🕨	₩.	Profile As	>
	*	Restore from Local History cppcheck Run C/C++ Code Analysis	>
		Team	>
		Configure	>
[Properties	Alt+Enter



② C/C++ Build > Settings を選択します。

sc Properties for MiV_SW									×
type filter text	Settings						\$	-	₩ 8
 Resource Builders C/C++ Build Build Variables 	Configuration:	Debug [Active]				 ✓ Manage 0 	Configura	tions	
Environment Logging	Tool Setting:	🖏 Toolchains 📕	Devices	Build Steps	P Build Artifact	Binary Parsers	8 Erro	r 🖣	•
Settings Tool Chain Editor > C/C++ General > cppcheclipse > MCU Project Natures Project References Run/Debug Settings	Target P Target P Target P Dotumiz Dotumiz Dotugg Solution Solution	rocessor ation s ing iC-V Cross Assemble ocessor des ellaneous iC-V Cross C Compi ocessor des mization ings ellaneous iC-V Cross C Linker	er	Script files	(T)	•	¥1 9 7	1 <u>b</u> 1	v
?						Apply and Close	Ca	ncel	

③ Configuration にて [All configurations] を選択します。

sc Properties for MiV_SW					—	×
type filter text	Settings				<-> -	> 🔻 🖇
 Resource Builders C/C++ Build Build Variables Environment Logging 	Configuration:	[All configurations] Debug [Active] Release [All configurations]		✓ Manag	ge Configuration ers 📀 Error 💽	∧ 15 ▶
Settings Tool Chain Editor	Target F Optimiz	Processor zation	Script files (-T)		🗐 🔊 🕲 🖓 🖗	1



④ Target Processor にて下記の通り設定します。

Architecture: RV32I (march=rv32i*)

Multiply extension: チェックを入れる

Integer ABI: ILP32 (-mabi=ilp32*)

sc Properties for MiV_SW			—	
type filter text	Settings		¢	• = = 8
 > Resource Builders ✓ C/C++ Build Build Variables 	Configuration: [All configurations]		 ✓ Manage Configu 	rations
Environment Logging Settings Tool Chain Editor	 Tool Settings Toolchains Devices Target Processor Optimization 	Build Steps 😤 Bu Architecture	uild Artifact 📓 Binary Parsers 🛛 Err RV32I (-march=rv32i*) on (RVM)	
> cppcheclipse > MCU	Warnings Debugging Solution Solu	Atomic extension Floating point	n (RVA) None	~
Project Natures Project References Run/Debug Settings	Preprocessor Includes	Compressed ext	ension (RVC) ILP32 (-mabi=ilp32*)	~
Run Debug Settings	Warnings Miscellaneous GNU RISC-V Cross C Compiler	Floating point ABI	None Teolohain dofault	~

⑤ GNU RISC-V Cross C Compiler を展開し Includes を選択します。

sc Properties for MiV_SW					×
type filter text	Settings		<	Þ ▼ ⇔ ·	• 8
 Resource Builders C/C++ Build Build Variables Environment 	Configuration: [All configurations]	~	Manage Config	urations	
Logging	Solution Settings Toolchains Devices	🎤 Build Steps 😤 Build Artifact 🗟 Binary Parsers	8 Error Parser	s	
Settings Tool Chain Editor > C/C++ General > cppcheclipse > MCU Project Natures Project References Run/Debug Settings	 Target Processor Optimization Warnings Debugging Solu RISC-V Cross Assembler Preprocessor Includes Warnings Miscellaneous 	Include paths (-I)	6 2 2) ∛I ∲I	
		Include system paths (-isystem)	6 1 2] 중 윤	



⑥ Include paths (-I) へ下記フォルダを追加します



						×			
								(২ া≣
Settings					() v	⇒ ♥ 8	isassembly	/ 🖸 Launch	n Conf
Configuration: [All configurations]				 ✓ Manage 	Configura	tions	des an outl	ine.	
🛞 Tool Settings 🛞 Toolchains 🔳 Devices 🎜	🖻 Build Steps 🧐	Build Artifact	Binary Parsers	8 Error Par	sers				
 Target Processor Optimization Warnings Debugging 	Include paths	(-1)		2	】 副 奇 Add	l ∲l			
 Solution State Solution State<td></td><td>sc Add direc</td><td>tory path</td><td></td><td></td><td></td><td></td><td></td><td>×</td>		sc Add direc	tory path						×
 Includes Warnings 		Directory:							
 Miscellaneous GNU RISC-V Cross C Compiler 									_
Preprocessor Preprocessor	Include system								
Optimization									
Warnings Miscellaneous			OK	Cancel		Workspa	ice	File system.	,
w S GNILL DISC V Cross C Linkor	11								

※ Ctrl キーで複数フォルダを一括選択、追加可能です。



⑦ Apply をクリックします。



⑧ Yes をクリックします。

sc Sett	ings ×
?	Changes made will not be reflected in the index until it is rebuilt. Do you wish to rebuild it now?
R	e <u>m</u> ember my decision
	<u>Y</u> es <u>N</u> o



⑨ GNU RISC V Cross C Linker > General を選択します。

🛞 Tool Settings 🛞 Toolchains 🔳 Devices 🎤	Build Steps P Build Artifact 🗟 Binary Parsers	8 Error Parsers
 Target Processor Optimization Warnings Debugging GNU RISC-V Cross Assembler Preprocessor Includes Warnings 	Script files (-T)	원 4월 28 등1 상1
Miscellaneous GNU RISC-V Cross C Compiler Preprocessor Includes Optimization Warnings Miscellaneous		
 Source State Stat		



⑩ リンカースクリプトを追加します。

Script files (-T) 欄の Add...ボタンより、miv_rv32_hal 下の miv-rv32-ram.ld 下記ファイルを追加します。

Settings						⇔ ⇒ ⇒ §	oisassembly O Launc	h C
Configuration: [All configurations]					✓ Manage C	onfigurations	des an outline.	
Tool Settings Stoolchains Devices	🎤 Build Steps	😤 Build Arti	fact 🗟 Bina	ary Parsers	O Error Pars	ers		
 Target Processor Optimization Warnings Debugging 	Script files	(-T)			2	11 월 집 월		
 SGNU RISC-V Cross Assembler Preprocessor Includes Warnings Miscellaneous GNU RISC-V Cross C Compiler Preprocessor Preprocessor 		sc Add file p File:	oath				>	<
Miscellappous			OK		Cancel	Workspace.	File system	
 Source and the second se								
C File selection		—		×				
Select one or more Workspace Files ✓	;			^				
> > drivers > > hal > > miv_rv32_hal								
miv_rv32_assert.h s miv_rv32_entry.S miv_rv32_hal_version.h								
i miv_rv32_hal.c i miv_rv32_hal.h								
 miv_rv32_init.c miv_rv32_plic.h miv_rv32_regs.h 								
miv_rv32_stubs.c miv_rv32_subsys.h miv_rv32_svscall.c								
miv-rv32-execute-in-p miv-rv32-ram.ld sample_fpga_design_c correject	lace.ld onfig.h			>				
?	OK		Cancel					



 Do not use standard start files にチェックを入れ、Apply and Close を クリックします。

🛞 Tool Settings 🛞 Toolchains 📕 Devices 🎤	Build Steps 🚇 Build Artifact 📓	Binary Parsers 8 Error Parsers
 Target Processor Optimization Warnings Debugging S GNU RISC-V Cross Assembler Preprocessor Includes Warnings Miscellaneous S GNU RISC-V Cross C Compiler Preprocessor Includes Optimization Warnings Miscellaneous S GNU RISC-V Cross C Linker General Eibraries Miscellaneous S GNU RISC-V Cross C Linker General B GNU RISC-V Cross C Linker S GNU RISC-V Cross C Linker S GNU RISC-V Cross Create Flash Image General S GNU RISC-V Cross Print Size 	Script files (-T)	ହି ହି ରୁ ଥି iv_rv32_hal/miv-rv32-ram.ld)*
	Do not use standard start files (Do not use default libraries (-noc No startup or default libs (-nos) Remove unused sections (-Xlink Print removed sections (-Xlinke Omit all symbol information (-s	-nostartfiles) odefaultlibs) tdlib) (ergc-sections) rprint-gc-sections) s) Restore Defaults Apply
		Apply and Close Cancel

12 Yes をクリックします。

sc Sett	ttings	×
?	Changes made will not be reflected in the index until it is now?	rebuilt. Do you wish to rebuild it
Re	Remember my decision	
		Yes <u>N</u> o



4-6. ソースコードの作成

① プロジェクトを右クリック > New > Source File をクリックします。

陷 Proje	ct Exp	lore	r 🗙 🕞 🔁 🍞 🕴				
🗸 🕵 M	iV_SW						
> 🛍	Inclu		New	>		Project	
> 🗁	, drive		Go Into		Ľ	File	
> 🗁	hal		Open in New Window		L,	File from Template	
> 🖻	, miv_		Show In	Alt+Shift+W >	Ċ	Folder	
			Show in Local Terminal	>	G	Class	
			Conv	Ctrl+C	h	Header File	
		in a	Daste	Ctrl+V	C	Source File	
		×	Delete	Delete	63	Source Fold	e file
			Source	>	C	C/C++ Project	
			Move			Other Ctrl+N	

② Source file にて main.c と入力し、Finish をクリックします。

sc New Sourc		(
Source File			C	
Create a new	source file.			
Source fol <u>d</u> er:	MiV_SW		<u>B</u> rowse	
Source fil <u>e</u> :	main.c			
<u>T</u> emplate:	Default C source template	\sim	Configure	
?	<u> </u>		Cancel	



③ 作成された main.c へ下記ソースコードを記入します。

```
main.c
/*
 * main.c
 */
#include "drivers/fpga_ip/CoreGPIO/core_gpio.h" /* GPIOを使うため */
#include "miv_rv32_hal/fpga_design_config.h"
#include "miv_rv32_hal/miv_rv32_hal.h"
#include "hal/hw_reg_access.h"
/* g_gpio_outを宣言。
* 構造体gpio_instance_tはcore_gpio.h内に定義されている。
* ここでCoreGPIOに関するデータを保持する。
*/
gpio_instance_t g_gpio_out;
/* LED点滅用 delay() */
void delay(int count)
{
   volatile int i;
   for (i = 0; i < count; i++);</pre>
}
int main()
{
   /****************
    * GPI0_init()
    *******************/
   /* GPIO_init()関数呼び出しによりCoreGPIO driverを初期化。
    * 他のGPIO driver関数を呼び出す前にGPIO_init()関数呼び出しが必要。
```



```
* 詳細はcore_gpio.hを参照。
   *
   * void GPIO_init
   * (
       gpio_instance_t * this_gpio,
               base_addr,
   *
        addr_t
   * gpio_apb_width_t bus_width
   * );
   * 第1引数:構造体gpio_instance_tへのポインター。今回はg_gpio_outと宣
≣。
   * 第2引数:初期化するGPIOのベースアドレス。
   * 第3引数: APBバス幅をdriverへ伝える。GPIO_APB_8_BITS_BUS or
GPIO_APB_16_BITS_BUS or GPIO_APB_32_BITS_BUS
   * 戻り値:なし
   */
  GPI0_init(&g_gpio_out, COREGPI0_OUT_BASE_ADDR,
GPIO_APB_32_BITS_BUS);
  * GPI0_set_output(), GPI0_set_outputs()
   /*
   * GPIO port 1つの出力値を設定したい場合はGPIO_set_output()、複数GPIO
port設定したい場合はGPIO_set_outputs()を使用
   * 詳細はcore_gpio.hを参照。
   *
   * void GPIO_set_outputs
   * (
   * gpio_instance_t * this_gpio,
   * uint32_t value
   * );
   *
   * void GPIO_set_output
   * (
```



```
*
       gpio_instance_t * this_gpio,
                      port_id,
    *
        gpio_id_t
         uint8_t
    *
                          value
    * );
    */
   /* 1 portずつ出力値を与える場合 */
   //GPI0_set_output( &g gpio_out, GPI0_0,1 );
   //GPI0_set_output( <u>&g gpio_out</u>, GPI0_1,0 );
   //GPI0_set_output( &g gpio_out, GPI0_2,1 );
   //GPI0_set_output( <u>&g gpio_out</u>, GPI0_3,0 );
   while (1)
   {
       /* GPIO_0~GPIO_3まで一括で出力値指定 */
       /* LED点灯 1010 */
       GPI0_set_outputs( &g_gpio_out, 0xA );
       delay(1000000);
       /* 点灯反転 0101 */
       GPI0_set_outputs( &g_gpio_out, 0x5 );
       delay(1000000);
   }
   return 0;
}
```

関数についての説明は、Driver のソースコード内、もしくは GitHub より 確認可能です。https://github.com/Mi-V-Soft-RISC-V





4-7. リンカスクリプトの編集

① miv-rv32-ram.ld ファイルを開きます。

Project Explorer ×	🖻 😫 🏹 🕴 🗖 🗖		
∽ 📂 MiV_SW			
> 🔊 Includes			
> 🗁 drivers			
> 🗁 hal			
🗸 🗁 miv_rv32_hal			
> 🔓 miv_rv32_assert.h			
> 📓 miv_rv32_entry.S			
> inv_rv32_hal_vers	ion.h		
> 🖻 miv_rv32_hal.c			
> 🔓 miv_rv32_hal.h			
> iniv_rv32_init.c			
> iniv_rv32_plic.h			
> inv_rv32_regs.h			
> 🖻 miv_rv32_stubs.c			
> h miv_rv32_subsys.h	1		
> iniv_rv32_syscall.c	:		
> 脑 sample_fpga_desi	ign_config.h		
miv-rv32-execu	New	>	
imiv-rv32-ram.ld	-		
> 🖻 main.c	Open		
	Show In	Alt+Shift+W >	
	Open With	>	Text Editor
	Show in Local Te	rminal >	System Editor
	🗎 Сору	Ctrl+C	In-Place Editor
[Paste	Ctrl+V	Default Editor
<	🗙 Delete	Delete	Other

ファイルについての説明は上部に記載されています。

	ww.
1/*************************************	TT
2 * Copyright 2019 Microchip FPGA Embedded Systems Solutions.	
3 *	
4 * SPDX-License-Identifier: MIT	
5 *	
6 * file name : miv-rv32-ram.ld	
7 * Mi-V soft processor linker script for creating a SoftConsole downloadable	
8 * debug image executing in SRAM.	
9 *	
10 * This linker script assumes that a RAM is connected at on Mi-V soft processo	r
11 * memory space pointed by the reset vector address.	
12 *	
13 * NOTE : Modify the memory section address and the size according to your	
14 * Libero design.	
15 * For example:	
16 * 1) If you want to download and step debug at a different RAM memory address	in
17 * your design (For example TCM base address) than the one provided in this	file.
18 * 2) The MIV_RV32, when used with MIV_ESS IP, provides ways to copy the execution of the minimum of the second	table
19 * HEX file from external Non-Volatile memory into the ICM at reset. In thi	s
20 * case your executable must be linked to the TCM address.	
22 * To know more about the memory map of the MIV_RV32 based Libero design, open	
23 " LITE MITY_KV32 IP CONTIGURATOR and 100K for Reset Vector Address and the	
24 memory map tab.	
25 · 26 */	
20 7	



② 33 行目にて、

アドレスが 0x80000000 になっていることを確認、 LENGTH を 16k へ変更し保存します。

```
30
31 MEMORY
32 {
33 ram (rwx) : ORIGIN = 0x80000000, LENGTH = 16k
34 }
35
```

Libero SoC、MIV_RV32_C0_0 での設定:

Configurator	—		\times
Mi-V RV32 Configurator			
Microsemi:MiV:MIV_RV32:3.1.200			
Configuration Memory Map			
AHB Initiator Address			
Start Address: Upper 16bits (Hex): 0x8000 Lower 16bits (Hex): 0x0			
End Address: Upper 16bits (Hex): 0x8fff Lower 16bits (Hex): 0xffff			
APB Initiator Address			
Start Address: Upper 16bits (Hex): 0x7000 Lower 16bits (Hex): 0x0	_		
End Address: Upper 16bits (Hex): 0x7fff Lower 16bits (Hex): 0xffff	_		
AXI Initiator Address			
Start Address: Upper 16bits (Hex): 0x6000 Lower 16bits (Hex): 0x0			
End Address: Upper 16bits (Hex): 0x6fff Lower 16bits (Hex): 0xffff	_		
TCM Address		1	
Start Address: Upper 16bits (Hex): 0x8000 Lower 16bits (Hex): 0x0	_		
End Address: Upper 16bits (Hex): 0x8000 Lower 16bits (Hex): 0x3fff			
TCM Access Support (TAS) Address		-	



4-8. fpga_design_config.h の用意

miv_rv32_hal フォルダ下のサンプル、sample_fpga_design_config.h を流用します。



- ① 名前を sample_fpga_design_config.h から fpga_design_config.h へ変更し、
 - ファイルを開きます。





② #define SYS_CLK_FREQ を 83333000UL へ変更します。

Libero SoC での CCC の Output Clock 設定:

Configurator

Clock Conditioning Circuitry (CCC)							
Actel:SgCore:PF_CCC:2.2.220							
Configuration PLL-Single							
Clock Options PLL Output Clocks							
For best results, put the highest frequency first.							
Output Clock 0							
l✓ Enabled							
Requested Frequency 83.333 MHz C Actual Lower 83.333 MHz							
Requested Phase 0 Degrees C Actual Lower 0 Deg							



③ GPIO のアドレスが 0x75000000UL となっていることを確認します。

	579) /************************************	***************************************				
	58	8 * Peripheral base addresses.					
	59	59 * Format of define is:					
	60	<pre>* <corename>_<instance>_BASE_ADDR</instance></corename></pre>					
	61	* The <instance> field is optional i</instance>	f there is only one instance of the core				
	62	62 * in the design					
	63	63 * MIV_ESS is an extended peripheral subsystem IP core with peripherals					
	64	4 * connections as defined below.					
	65	* The system can be further extended	by attaching APB peripherals to the				
	66	<pre>* empty APB slots.</pre>					
	67	*/					
	68	<pre>#define MIV_ESS_PLIC_BASE_ADDR</pre>	0x70000000UL				
	69	<pre>#define COREUARTAPB0_BASE_ADDR</pre>	0x71000000UL				
	70	<pre>#define MIV_MTIMER_BASE_ADDR</pre>	0x72000000UL				
	71	<pre>#define MIV_ESS_APBSLOT3_BASE_ADDR</pre>	0x73000000UL				
ł	72	#define MIV ESS APBSLOT4 BASE ADDR	0x74000000UL				
l	73	<pre>#define COREGPIO_OUT_BASE_ADDR</pre>	0x75000000UL				
	74	<pre>#define CORESPI_BASE_ADDR</pre>	0x76000000L				
	75	<pre>#define MIV_ESS_uDMA_BASE_ADDR</pre>	0x78000000L				
	76	<pre>#define MIV_ESS_WDOG_BASE_ADDR</pre>	0x79000000UL				
	77	<pre>#define MIV_ESS_I2C_BASE_ADDR</pre>	0x7A000000UL				
	78	<pre>#define MIV_ESS_APBSLOTB_BASE_ADDR</pre>	0x7B000000UL				
	79	<pre>#define MIV_ESS_APBSLOTC_BASE_ADDR</pre>	0x7C000000UL				
	80	<pre>#define MIV_ESS_APBSLOTD_BASE_ADDR</pre>	0x7D000000UL				
	81	<pre>#define MIV_ESS_APBSLOTE_BASE_ADDR</pre>	0x7E000000L				
	82	#define MIV_ESS_APBSLOTF_BASE_ADDR	0x7F000000L				
	83						

Libero SoC にて、今回は APB Initiator Address を 0x70000000~とし、 MIV_ESS を接続しています。

Configurator	-	×
Mi-V RV32 Configurator		
Microsemi:MiV:MIV_RV32:3.1.200		
Configuration Memory Map		
AHB Initiator Address		
Start Address: Upper 1 6 bits (Hex): 0x8000 Lower 1 6 bits (Hex): 0x0		
End Address: Upper 16bits (Hex): 0x8fff Lower 16bits (Hex): 0xffff		
APB Initiator Address		
Start Address: Upper 1 6 bits (Hex): 0x7000 Lower 1 6 bits (Hex): 0x0		
End Address: Upper 1 6 bits (Hex): 0x7fff Lower 1 6 bits (Hex): 0xffff		
AXI Initiator Address		



GPIO の offset address は MIV_ESS User Guide > Table 2-2. Peripheral Module Address Offsets から確認可能です。 https://ww1.microchip.com/downloads/aemDocuments/documents/FPGA/ProductDo cuments/UserGuides/ip_cores/directcores/MIVESS.pdf#page=13

Base_Address_of ESS + Offset address of GPIO = CoreGPIO address 0x7000 0000 + 0x0500 0000 = 0x7500 0000

4-9. #include パスの修正

プロジェクトに Import した GitHub のファイル内には、fpga_design_config.h を 呼び出しているものがあります。 現在の構成に合わせてパス表記を変更します。

miv_rv32_hal 内 miv_rv32_hal.h ファイルを開き #include " fpga_design_config /fpga_design_config.h"を #include "fpga_design_config.h"へ変更





4-10. ビルド

プロジェクトを右クリックし、Build Project からビルドします。



5. 実行

- ① 開発キットが PC に接続されていることを確認します。
- ② プロジェクトを右クリックし、Debug As > Debug Configurations...をクリックします。

Project Exp	lore	r 🗙 🕞 🔁 🖓 🖇 🗖 🗖			
🗸 🍰 MiV_SW	1				
> 🖑 Bina		New	>		
> 🗊 Inclu		Go Into			
> 🧽 Deb > 🗁 drive > 🗁 hal ~ 📂 miv_)eb Irive nal niv_	Open in New Window Show In Show in Local Terminal	Alt+Shift+W > >		
> 🖻 fr		Сору	Ctrl+C		
> 🖻 m	ß	Paste	Ctrl+V		
> 🔝 m	×	Delete	Delete		
> 🖻 m		Source	>		
> <u>ke</u> m > lini m		Rename	F2		
> 🖬 m	24 24	Import Export			
> 🖻 m	٢	Robot Framework	>		
n 100 < 100 k 100	Ł	Build Project Clean Project Refresh Close Project	F5		
> 🖻 mair		Close Unrelated Project			
		Build Configurations	>		
		Build Targets	>		
		Index	>		
		Profiling Tools	>		
🎋 Debug 🗡	0	Run As	>	-	
× -	*	Debug As	>		1 Robot Test
		Profile As	>		2 KODOT KEMOTE lest
		Restore from Local History		E.	
		cppcneck	>		Debug Configurations


③ GDB OpenOCD Debugging をダブルクリックします。

sc Debug Configurations	- D X
Create, manage, and run configurat	ions
Image: Second system Image: Second system Image: Secon	 Configure launch settings from this dialog: Press the 'New Configuration'figuration of the selected type. Press the 'New Prototype' butt prototype of the selected type. Press the 'Export' button to export the selected configurations. Press the 'Duplicate' button to copy the selected configuration. Press the 'Delete' button to remove the selected configuration. \$\vee\$ Press the 'Filter' button to configure filtering options. Edit or view an existing configuration by selecting it. Select launch configuration(s)' menu item to link a prototype. Select launch configuration(s) to reset with prototype values.
Filter matched 2 of 9 items	
0	Debug Close

④ Debugger タブを開き、Start OpenOCD locally ヘチェックを入れ
 Apply 後、Debug をクリックします。

sc Debug Configurations		_		×
Create, manage, and run	configurations			Ť
Image: Second system Image: Second system v Image: Second system Image: Second system Image: Second system Image: Second system	Name: MiV_SW Debug Main Debugger Startup Source ⊆ommon SVD Path OpenOCD Setup Start OpenOCD locally Executable path: \$(openocd_path)/\$(openocd_executable}) Actual executable: C:¥Microchip¥SoftConsole-v2022.2-RISC-V-747¥eclipse¥//oper (to change it use the global or workspace preferences pages or the global or workspace page	Bro ocd/bir he proje	owse //openc ect prop	Varia poct.ex poerties
	Revert		Арр	οly
?	Debug		Clo	ose



⑤ ▶の Resume ボタンをクリックします。



- ⑥ Discovery Kit にて LED の点滅を確認します。
- ⑦ SoftConsole にて Debug 実行を止めます。

్	SoftConsole からの Debug 実行と、Libero SoC からの書き込みは
	同時にできません。
	必要に応じて適時 Debug 実行は停止して下さい。

SoftConsole - MiV_SW/main.c - SoftConsole v -RISC-V-File Edit Source Refactor Navigate Search Project Git Run Window Help MiV_SW Debug 🗸 🔅 🔄 🚳 🖛 🖄 🖉 🕬 💷 💠 🔳 🔅 Debug 🖻 😒 🎖 🕴 🗖 🗈 main.c × 🕩 miv_rv32_hal.h 陷 Project Explorer 🗵 00 gpio_instance_t **LILLS** 🗸 😂 MiV_SW * 39 addr t base

SoftConsole の Debug を止めずに、Libero SoC にて Run PROGRAM Action を 行った場合:

Error	×
No programmer is c	onnected.
ОК	



- 6. ソフトウェアをデバイスへ書き込んでみよう
- ※ ここでは一例として TCM の initialize ファイルとして割り当てます。
- ※ 例えば PolarFire SRAM (AHBLite and AXI)ブロックを使っている場合は 同様の方法で SRAM の initialize ファイルとして割り当てます。
- ※ Discovery Kit 上には SPI Flash が搭載されていないため、SW を SPI Flash へ 書き込むことはできません。
- SoftConsole にて、ビルド後 Debug フォルダ下に MiV_SW.hex が生成されていることを 確認します。



② 任意のテキストエディタで hex ファイルを開きます。

HiV_S	W.hex 🗵
1	-: 020000480007A
2	:100000006F00101A6F00C00800000000000000000000000000000000
3	:100010006F00001100000000000000000000000
4	:100020006F00801800000000000000000000000000000
5	:100030006F0000200000000000000000000000000
6	:1000400000000006F0040276F00802F00000000BC
7	:10005000000000000000000000000000000000
8	:100060006F0080696F0040366F00803E6F00C046B1
9	:100070006F00004F6F0040576F00805F6F0040704F
10	:100080006F0080781300000013000001300000000
11	:10009000130101F823201100232221002324310021
12	:1000A0002326410023285100232A6100232C7100BC
13	:1000B000232E8100232091022322A1022324B102B6
14	:1000C0002326C1022328D102232AE102232CF10294
15	:1000D000232E01032320110523222105232431058A
16	:1000E0002326410523285105232A6105232C710568
17	:1000F000232E8105232091072322A1072324B10762
18	:100100002326C1072328D107232AE107232CF1073F
19	:1001100073252034F3251034EF00504A6F00C07F60
20	-100100001201010022001100220021002204210000



③1行目を削除し、保存します。

HiV_S	W.hex 🗵
1	:10000006F00101A6F00C00800000000000000000000000000000000
2	:100010006F00001100000000000000000000000
3	:100020006F00801800000000000000000000000000000
4	:100030006F0000200000000000000000000000000
5	:1000400000000006F0040276F00802F00000000BC
6	:10005000000000000000000000000000000000
7	.1000600060000606000 0 006600000000000000

④ Libero SoC にて、Design Flow タブ内で

Configure Design Initialization Data and Memories をダブルクリックします。

Project File Edit View Design Tools Help
□ □
Design Flow 🗗 🗙 Top Module(root): miv_top 🖸 🖸 💕
Top Module(root): miv_top 🛛 D 🖬 🖉 🌮
Top Module(root): miv_top E 🖸 🚺 🕼 🜮
A still of Orientities also Texade are established as an estimate
Active Synthesis Implementation: synthesis
Tool
🗏 💩 Open SmartTime
🗌 🖹 Verify Power
🗆 🛱 Open SSN Analyzer
🖻 🕨 Configure Hardware
Programming Connectivity and Interface
- 🖓 Configure Programmer
- 👵 Select Programmer
🖻 🕨 Program Design
✓ Generate FPGA Array Data
-• Configure Design Initialization Data and Memories
✓ Generate Design Initialization Data
Example 2 Configure I/O States During JTAG Programming
Configure Programming Options
Configure Security
V Generate Bitstream
Configure Actions and Procedures
V Run PROGRAM Action
Program SPI Flash Image
Generate SPI Flash Image
Design Flow Design Hierarchy Stimulus Hierarchy Catalog Components Files

O Libero - C:¥miv_lab¥miv_top¥miv_top.prjx



⑤ Fabric RAMs のタブを開きます。

Logical Instance Name から TCM を見つけ、右クリック、Edit...をクリックします。

Reports 5 × Constraint Manager 5 Design Initialization UPROM SNVM SPI Flas	× StartPage & × Design and Memory Initialization	
Apply Discard Help Usage statistics LSRAM Memory Available Memory(Bytes): 788480	Clients Load design configuration Edit Initialize all clients from: Initialize all Clients from sNVM Filter out Inferred RAMs	
Used Memory(Bytes): 17920 Free Memory(Bytes): 770560	Logical Instance Name	Dep
	1 MIV_RV32_C0_0/MIV_RV32_C0_0/u_ipcore_0/gen_tcm0.u_subsys_TCM_0/tcm_ram.u_ram_0/mem[31:0] Edit	4096
	2 MIV_RV32_C0_0/MIV_RV32_C0_0/u_ipcore_0/u_hart_0/u_expipe_0/gen_gpr_ram.u_gpr_0/gen_gpr.u_gpr_array_0/mem_xf[31:0]	32x3
	3 MIV_RV32_C0_0/MIV_RV32_C0_0/u_ipcore_0/u_hart_0/u_expipe_0/gen_gpr_ram.u_gpr_0/gen_gpr.u_gpr_array_0/mem_xf_1[31:0]	32x3
Used space	4 MIV_RV32_C0_0/MIV_RV32_C0_0/u_ipcore_0/u_subsys_interconnect_0/u_subsys_regs/u_req_buffer/gen_buff_loop[0].buff_data[6:0)] 2x6

⑥ Content from file を選択し、...ボタンから MiV_SW.hex を選択します。
 Storage Type はデフォルトの sNVM のままとします。

🕑 Edit Fabric RAN	V Initialization Client			?	×
Olient name: V_F	?V32_CO_O/u_ipcore_O/gen_tcmO.u_subsy	/s_TCM_C)/tcm_ram.u_ram_0/	'mem[31 :	D]
Physical Name: 🔽)/gen_tcm0.u_subsys_TCM_0/tcm_ram.u_v	ram_0/me	em_mem_0_5/INST_	RAM1 K2	D_IP
RAM Initializatio	n Options				
C Initialized Co	ntent from Synthesis				
C Content Initia	alized from configurator(using content e	ditor opti	on)		_
 Content from 	n file: C:/miv_lab/miv_top/SoftConsole/	/MiV_SW	/Debug/MiV_SW.he:	×	
C Content filled C No content (d with Os client is a placeholder and will not be pr	ogramme	d)		
Optimize for:	ligh Speed € Low power √M _		ОК	Close	
Import Memory Fil	ie		?	×	
Look in: 📜 C:¥r	niv_lab¥miv_top¥SoftConsole¥MiV_SW¥Debug	• (3 0 0 🖗 🖽		
S My Computer	Name	Size	Туре		
S My Computer	Name	Size	Type File Folder File Folder		
S My Computer	Name drivers miv_rv32_hal hal	Size	Type File Folder File Folder File Folder		
Jy Computer 21588	Name drivers miv_rv32_hal hal MiV_SW.hex	Size	Type File Folder File Folder File Folder B hex File		
My Computer State 21588 File name: MiV_SW.	Name I drivers I miv_rv32_hal hal MiV_SW.hex I hex	Size	Type File Folder File Folder File Folder B hex File	►	
J My Computer 21588 File name: MiV_SW. Files of type: Intel-He	Name drivers miv_rv32_hal hal MiV_SW.hex	16.29 K	Type File Folder File Folder File Folder B hex File Qpen Cancel		
My Computer My Computer Cartery Computer Cartery Computer Miv_SW. Files of type: Intel-He Cuse absolute path	Name drivers miv_rv32_hal hal MiV_SW.hex hex x(*.hex *.ihx) (file will not be copied if you move the design)	16.29 K	Type File Folder File Folder B hex File Qpen Cancel		
My Computer 21588 File name: MiV_SW. Files of type: Intel-He Use absolute path 1 Use relative path 1	Name drivers miv_rv32_hal hal MiV_SW.hex (hex ex(*.hex *.ihx) file will not be copied if you move the design)	16.29 K	Type File Folder File Folder B hex File Qpen Cancel	•	



 ⑦ Initialize all clients from が Initialize all Clients from sNVM となっていることを 確認します。

Clients				
Load design configuration Edit	Initialize all clients from: Initialize all Clients from sNVM	<u> </u>		
🗌 Filter out Inferred RAMs				
Logical Instance Nat	ne	PORTA Depth * Width	PORTB Depth * Width	Memor
1 /u_ipcore_0/gen_tcm0.u_subsys_TCM_0/tcm_ram.u_ram_0/mem[31:0]			4096x32	C:/miv_l
2 /u_ipcore_0/u_hart_0/u_expipe_0/gen_gpr_ram.u_gpr_0/gen_gpr.u_gpr_array_0/mem_xf[31:0]			32x32	No cont
	0/ 0/ / //DA 00	22.22	22.22	

⑧ Apply します。

Design and Memory Initialization* 🗗 🗙 🛛 Re	ports 🗗 × 📔 Constraint Manager 🗗 × 🗍 StartPage
Design Initialization uPROM sNVM SPI Flash	n Fabric RAMs≭ eNVM
Apply Discard Help	
Usage statistics	Clients
LSRAM Memory	Load design configuration Edit Initialize a
Available Memory(Bytes): 788480	Filter out Inferred RAMs
Free Memory(Bytes): 770560	Logical Instance Name

⑨ Generate Design Initialization Data をダブルクリックします。





Run PROGRAM Action をダブルクリックし書き込みます。



 Discovery Kit のケーブルを抜き差しし SW が書き込まれていること (SoftConsole からの Debug 実行なしで LED が点滅すること)を確認します。



7. UART を動かしてみよう

既存のLチカのプロジェクトを流用し Hello World を表示させます。

7-1. ハードウェア

① Libero SoC にて MIV_ESS_C0_0 ブロックを開きます。





② General タブにて UART ヘチェックを入れます。

O Configurator	<
MIV_ESS Actel:SystemBuilder:MIV_ESS:2.0.200	
Image: Constrant of the second strant of	
Apply New preset]
Help	

UART の詳細設定は UART タブにて可能です。

今回はデフォルトのままとします。

Configurator			- 0	×
MIV ESS				
Actel:SystemBuilder:MIV_ES	SS:2.0.200			
	General Boots	strap APB 🛛 uDMA GPIO PLIC SPI 🚺 Time	r 🚯 UART	-
MIV_ESS_UI_default_co	Core Configuration			-
DGC1_PF_SPI_BOOT DGC2_PF_I2C_BOOT	TX FIFO:	Disable TX FIFO 💌		
DGC3_PF_uPROM_BOOT	RX FIFO:	Disable RX FIFO 👤		
	Configuration:	Programmable		
	Baud Value:	1		
	Character Size:	7 bits		
	Parity:	Parity Disabled		
	RX Legacy Mode:	Disabled 👤		
Apply New preset	FIFO Implementation	: In RAM		
	Status Flags:	6		
	Baud Value Precision —			_
	Enable Extra Precisio	in: 🗖		
	Fractional Part of Ba	ud Value: +0.0 <u>*</u>		-
	•			
Help 🔻			ОК Са	ncel



③ MIV_ESS_C0_0 ブロックを右クリック、Update Instance をクリックし ブロックをアップデートします。



UART_RX、UART_TXのピンが表示されます。



④ Promote to Top Level を使用し、UART_RX、UART_TX のポートを出します。





⑤ Smart Design を保存し、Generate Component をクリックします。



⑥ Build Hierarchy をクリックします。



 ⑦ Constraint Manager を開き、Timing タブにて Derive Constraints をクリックし、 sdc ファイルを再生成します。

Reports & StartPage & Con	nstraint Manager 🛛 🗗	× Design and	Memory Initia
I/O Attributes Timing Floor Planner Netlist	Attributes		
New I Import Link	Edit 🔽 Che ol	k 🗣 \Lambda Derive Co	nstraints
	Synthesis	Place and Route	Timing Ver
constraint¥miv_top_derived_constraints.sdc	\checkmark		✓

⑧論理合成を行います。





 ⑨ Constraint Manager より I/O editor を開き、UART_RX、UART_TX を ピンアサインします。
 UART_RX: W21
 UART_TX: Y21

Pir	n View 🗗	Port View [a	otive] 🗗	XC	VR View 🗗 🛛	Mem	ory View 🗗 🗎	IOI) View 🗗	Package View 🗗		Floorplanr
	Port	Name 🚺	Directio	in 🔽	I/O Standa	rd 💌	Pin Number	•	Locked	Macro Cell	•	Bank Name
1	🔻 GPIC	_OUT										
2	GF	PIO_OUT[0]			LVCMOS	18	T18		✓			Bank0
3	GF	PIO_OUT[1]			LVCMOS	18	V17		✓			Bank0
4	GF	PIO_OUT[2]			LVCMOS	18	U20		✓			Bank0
5	GF	PIO_OUT[3]			LVCMOS	18	U21		✓			Bank0
6	REF_	CLK_0			LVCMOS	18	R18		✓			Bank0
7	TCK		INPU	Т			A8		✓	UJTAG_SEC		
8	TDI		INPU	Т			A9		✓	UJTAG_SEC		
9	TDO		OUTPL	JT			A7		✓	UJTAG_SEC		
10	TMS		INPU	т			B7		✓	UJTAG_SEC		
11	TRST	В	INPU	т			B9		✓	UJTAG_SEC		
12	UAR	T_RX			LVCMOS	18	W21		✓			Bank0
13	UAR	T_TX			LVCMOS	18	Y21		✓			Bank0

UART_RX、UART_TXのピン番号は、Discovery Kitの回路図より確認可能です。 https://www.microchip.com/en-us/development-tool/mpfs-disco-kit

- 10 Place and Route(配置配線)を行います。
- ① Generate FPGA Array Data をダブルクリックします。

(Configure Design Initialization Data and Memories を開くため事前に実行が必要。)





⑫ Configure Design Initialization Data and Memories を開きます。

	÷
÷	Program Design
	Generate FPGA Array Data
	-• Configure Design Initialization Data and Memories
	Generate Design Initialization Data
	🔚 Configure I/O States During JTAG Programming
	Configure Programming Options
	🛛 🔞 Configure Security

¹³ L チカの hex ファイルが設定されているので削除します。

	TCM を右クリック	、Edit で開き、	No content	を選択します
--	------------	------------	------------	--------

Reports 🗗 × 🛛 Constraint Manager 🗗 ×	StartPage & × 😣 Design and Memory Initialization & × 💁 miv_top & ×
Design Initialization uPROM sNVM SPI Flash	8 Fabric RAMs eNVM
Apply Discard Help	-Ollents
LSRAM Memory Available Memory(Bytes): 788480 Used Memory(Bytes): 17920	Load design configuration Edit Initialize all clients from: Initialize all Clients from sNVM Imitialize all Clients from sNVM Imitialize all Clients from sNVM
Free Memory(Bytes): 770560	Logical Instance Name
	1 3 MIV_RV32_C0_0/MIV_RV32_C0_0/u_ipcore_0/gen_tcm0.u_subsys_TCM_0/tcm_ram.u_ram_0/mem[31:0] Edit
	2 MIV_RV32_C0_0/MIV_RV32_C0_0/u_ipcore_0/u_hart_0/u_expipe_0/gen_gpr_ram.u_gpr_0/gen_gpr.u_gpr_array_0/mem_xf[:
	3 MIV_RV32_C0_0/MIV_RV32_C0_0/u_ipcore_0/u_hart_0/u_expipe_0/gen_gpr_ram.u_gpr_0/gen_gpr.u_gpr_array_0/mem_xf_
Used space	4 MIV_RV32_C0_0/MIV_RV32_C0_0/u_ipcore_0/u_subsys_interconnect_0/u_subsys_regs/u_req_buffer/gen_buff_loop[0].buff_

Contemporary Edit Fabric RAM Initialization Client	\times
Client name:ipcore_0/gen_tcm0.u_subsys_TCM_0/tcm_ram.u_ram_0/mem[31:0]	
Physical Name: subsys_TCM_0/tcm_ram.u_ram_0/mem_mem_0_5/INST_RAM1K20_IA	2
RAM Initialization Options	
C Initialized Content from Synthesis	
C Content Initialized from configurator(using content editor option)	
C Content from file:	
C Content filled with Os	
No content (client is a placeholder and will not be programmed)	
Optimize for: 🖲 High Speed 🖸 Low power	
Storage Type SNVM	
Help OK Close	

(④ Run PROGRAM Action にて書き込みます。



7-2. ソフトウェア

① SoftConsole にて、GitHub からダウンロード、解凍した Driver から CoreUARTapb を追加でインポートします。





インポート後:



② プロジェクトのプロパティにて、Include Path へ CoreUARTapb を追加します。





sc Folder selection	—		×
Select one or more Worksp	ace Folo	ders	
🗸 😂 MiV_SW			
> 🗁 .settings			
> 🗁 Debug			
🗸 🗁 drivers			
🗸 🗁 fpga_ip			
> 🗁 CoreGPIO			
> 🗁 CoreUARTa	pb		
> 🗁 hal			
🗁 miv_rv32_hal			
OK		Cance	I

② main.c として下記ソースコードを記載します。

main.c
/*
* main.c
*/
<pre>#include "drivers/fpga_ip/CoreGPIO/core_gpio.h"</pre>
<pre>#include "miv_rv32_hal/fpga_design_config.h"</pre>
<pre>#include "miv_rv32_hal/miv_rv32_hal.h"</pre>
#include "drivers/fpga_ip/CoreUARTapb/core_uart_apb.h" /* UARTを使うため */
<pre>#include "hal/hw_reg_access.h"</pre>
/**************************************
* Instruction message. This message will be transmitted over the UART to
* HyperTerminal when the program starts.

<pre>uint8_t testmsg[] = {"Hello World!"};</pre>
/*
* UART instance data.
*/
UART_instance_t g_uart;





③ プロジェクトをビルドします。







④ Debug Configurations...よりデバッグを開始します。



⑤ Tera Term 等、任意の Terminal を開きます。
 ポートごとに複数 Terminal を起動します。
 ※ ポート番号は環境により異なります。

💆 Tera 1				
ファイル(F)	編集(E)	設定(S)	コントロール(O)	ウィンドウ(\
		端末	₹(T)	
		ウイン	ッドウ(W)	
		フォン	ット(F)	>
		+-7	ポード(K)	
		シリ	アルポート(E)	
		プロ・	キシ(P)	

Tera Term: シリアルポート 設定と接続





⑥ シリアル通信の設定を確認、適宜変更します。

Tera Term: シリアルポート 設定と接続

ポート(P):	COM67	\sim
スピード(E):	115200	~
データ(D):	8 bit	\sim
パリティ(A):	none	\sim
ストップビット(S):	1 bit	\sim
フロー制御(F):	none	\sim

⑦ Resume ボタンを押して、アプリケーションを実行します。

sc SoftConsole - MiV_SW/main.c - SoftConsole v2022.2-RISC-V-747

File Edit Source Refactor Navigate Search Project Git Run Window Help

🐐 🔳 🕸 Debug	∽ ViV_SW Debug	× 🔅	6 4	• 🔍 🛛		• 3	R
🔁 Project Explorer 🗙	🖻 🔄 🍸 🕴 🗖 🚺 main.c 🗙				Resur	ne (F8	3)

⑧ Terminal にて Hello World!が表示されることを確認します。





何も表示されない場合、UART の RX、TX のピン配置が逆になっていないか 確認します。 表示されるものの文字化けする場合は Terminal のボーレート設定を変えることで、 改善されるか確認します。ボーレート変更で改善する場合、CCC などクロック周りの 設定を見直してみます。

8. 関連ドキュメント

詳細についてはメーカのドキュメントをご参照下さい。

 PolarFire® SoC Discovery Kit https://www.microchip.com/en-us/development-tool/mpfs-disco-kit

• AN4997: PolarFire FPGA Building a Mi-V Processor Subsystem Application Note (Earlier TU0775)

https://www.microchip.com/en-us/application-notes/an4997

- GitHub Mi-V Soft RISC-V
 <u>https://github.com/Mi-V-Soft-RISC-V</u>
- MIV RV32

https://www.microchip.com/en-us/products/fpgas-and-plds/ip-core-tools/miv-rv32

• MIV_ESS

https://www.microchip.com/en-us/products/fpgas-and-plds/ip-core-tools/miv-ess

以上



改版履歴

リビジョン	日付	概要
V1	2025年3月	新規作成

免責およびご利用上の注意

1. 弊社より資料を入手されましたお客様におかれましては、下記の使用上の注意を一読いただいた上でご使用ください。

- 2. 本資料は予告なく変更することがあります。
- 本資料の作成には万全を期していますが、万一ご不明な点や誤り、記載漏れなどお気づきの点がありましたら、
 本資料を入手されました下記代理店までご一報いただければ幸いです。

株式会社マクニカ ホームページ https://www.macnica.co.jp/

- 本資料で取り扱っている回路、技術、プログラムに関して運用した結果の影響については、責任を負いかねますので あらかじめご了承ください。
- 5. 本資料は製品を利用する際の補助的な資料です。製品をご使用になる際は、各メーカ発行の英語版の資料も あわせてご利用ください。

