PolarFire SoC ベアメタルデザインの実行

PolarFire SoC Icicle Kit 版



内容

はじめに	3
Microchip 社の SW ソリューションについて	3
環境	3
1. 事前準備	4
1-2. Icicle Kit のジャンパー・ピンの確認	4
1-2. Silicon Labs CP2108 drivers のインストール	4
1-3. Icicle Kit と PC の接続	4
1-4. SoftConsole の環境変数設定	5
2. Libero SoC プロジェクトの用意	6
2-1. PolarFire SoC Icicle Kit Reference Design のダウンロード	6
2-2. tcl スクリプトの実行	8
2-3. 書き込み	12
3. SoftConsole を用いたベアメタルアプリケーションの実行	15
3-1. PolarFire SoC Bare Metal Examples のダウンロード	15
3-2. SoftConsole の起動	16
3-3. RTC-time exmple のインポート	17
3-4. ビルド前の編集	20
3-5. Build Configurations の選択	22
3-6. ビルド	22
3-7. Terminal の立ち上げ	24
3-8. Icicle Kit の電源投入	25
3-9. デバッグ実行	



はじめに

この資料は Microchip 社の PolarFire SoC をはじめて触る人向けに ベアメタルの動作確認の一通りの操作フローを記載した資料です。

Microchip 社の SW ソリューションについて

・ARM Cortex-M3 がハード的に搭載された SmartFusion2
・ソフト CPU として RISC-V を載せる Mi-V
・RISC-V がハード的に搭載された PolarFire SoC
といった選択肢があります。
お客様のアプリケーションに応じて選択下さい。
本資料は PolarFire SoC を対象としています。

環境

PolarFire SoC Icicle Kit (MPFS-ICICLE-KIT-ES) Libero SoC v2024.2 SoftConsole v2022.2 icicle-kit-reference-design v2024.09 polarfire-soc-bare-metal-examples v2024.05



1. 事前準備

1-2. lcicle Kit のジャンパー・ピンの確認

ジャンパー・ピンの設定を Default へ設定。

https://github.com/polarfire-soc/polarfire-soc-documentation/blob/master/reference-designs-fpga-and-development-kits/icicle-kit-user-guide.md#jumpers

1-2. Silicon Labs CP2108 drivers のインストール

https://www.silabs.com/developers/usb-to-uart-bridge-vcp-drivers

PolarFire SoC Icicle Kit User Guide > 3.9.1 USB to UART Interface の項目を参照 https://ww1.microchip.com/downloads/aemDocuments/documents/FPGA/ProductDo cuments/UserGuides/microchip_polarfire_soc_fpga_icicle_kit_user_guide_vb.pdf#page=14

1-3. lcicle Kit と PC の接続

USB Embedded Programming connector は J33、USB-UART Terminal は J11 です。 IcicleKit と PC を接続し、Power ON/OFF Switch を入れます。





1-4. SoftConsole の環境変数設定

LiberoSoC のインストールパスに応じて

ファイル内で環境変数を設定。



書き方については softconsole.cmd の冒頭コメントを参考にして下さい。

3	REM	PolarFireSoC boot mode programming
4	REM	For PolarFireSoC boot mode programming using mpfsBootmodeProgrammer.jar the fpgenprog
5	REM	tool must be installed. Fpgenprog is shipped with 'Libero' and with 'Program and Debug'
6	REM	tools and SoftConsole must know where it is installed.
7	REM	
8	REM	The mpfsBootmodeProgrammer.jar will try to autodetect the fpgenprog tools if it's
9	REM	installed in the default locations, more details about that here:
10	REM	https://mi-v-ecosystem.github.io/SoftConsole-Documentation/SoftConsole-v2022.2/using softconsole/mpfs.html
11	REM	
12	REM	However, if fpgenprog path can't be autodetected then it must be set by the user.
13	REM	Either by setting it in SoftConsole External Tool launch configuration's Environment
14	REM	tab for each launcher. Or once globally by setting it here by:
15	REM	set FPGENPROG= <full absolute="" binary="" fpgenprog="" path="" to=""></full>
16	REM	
17	REM	Example (full path including the binary filename):
18	REM	set FPGENPROG=C:\Microsemi\Libero SoC v2022.2\Designer\bin64\fpgenprog.exe
19	REM	
20	REM	·
21		
22		
23	REM	JDC Helper
24	REM	For jdchelper jar the jdcgen tool must be installed. Jdcgen is shipped with:
25	REM	'Libero' and with 'Program and Debug' tools and SoftConsole must know where it is installed.
26	REM	
27	REM	The jdcHelper.jar will try to autodetect the jdcgen tools if it's
28	REM	installed in the default locations, more details about that here:
29	REM	https://mi-v-ecosystem.github.io/SoftConsole-Documentation/SoftConsole-v2022.2/using softconsole/mpfs.html
30	REM	
31	REM	However, if jdcgen path can't be autodetected then it must be set by the user.
32	REM	Either by setting it in SoftConsole External Tool launch configuration's Environment
33	REM	tab for each launcher. Or once globally by setting it here by:
34	REM	set JDCGEN= <full absolute="" binary="" fpgenprog="" path="" to=""></full>
35	REM	
36	REM	Example (full path including the binary filename):
37	REM	set JDCGEN=C:\Microsemi\Libero SoC v2022.2\Designer\bin\jdcgen.exe
38	REM	
39	REM	

設定しない場合エラーが出ることがあります。 https://forum.microchip.com/s/topic/a5C3l000000Md59EAC/t380286



2. Libero SoC プロジェクトの用意

2-1. PolarFire SoC Icicle Kit Reference Design のダウンロード

Icicle Kit 向けの Libero SoC リファレンスデザインは tcl スクリプトとして 提供されています。 tcl スクリプトは GitHub から入手します。tcl スクリプトから Libero SoC プロジェクト生成する手順については GitHub 上に記載されています。 https://github.com/polarfire-soc/icicle-kit-reference-design

ここでは例として現在の最新版 v2024.09 を使用します。 右側の v2024.09 をクリックします。

polarfire-soc / icicle-kit-reference-d	icle-kit-reference-design (Public)		
<> Code ⊙ Issues 5 \$1 Pull requests (🕞 Actions 🖽 Projects 😲 Security 🗠 Insights		
ያ master 👻 የ 1 Branch 🛇 20 Tags	Q. Go to file	<> Code -	About
Brian Burke and Brian Burke PCI class code: U	lpdated to ensure 'Ispci -v' command rep 🚥 d67b45a · 5 days ago	🕓 510 Commits	PolarFire SoC Icicle Kit Libero reference design
🖿 .ci	Added smarthls entry to .ci file	2 years ago	🗘 Readme
github	github: update broken link in contribution docs	last year	
🖿 XML	MSS config: update LPDDR4 tRFC value	11 months ago	- Activity
📄 diagrams	Block diagram: fix typo	2 years ago	Custom properties
script_support	PCI class code: Updated to ensure 'Ispci -v' command report	5 days ago	 39 watching
🗅 .gitignore	Merge branch 'develop' of ssh://bitbucket.microchip.com/fp	2 years ago	약 14 forks
LICENSE.md	License: update name in license	last year	Report repository
MPFS_ICICLE_KIT_REFERENCE_DESIGN.tcl	Update COREAXI4DMACONTROLLER and LSRAM version	3 months ago	Releases 19
🗅 Readme.md	readme: add gpios used by the opcua demo	last year	v2024.09 Latest on Oct 11
다 README 《 Code of conduct 최 Licens	e	:=	+ 18 releases
			Packages

PolarFire® SoC Icicle Kit Reference Design Generation

No packages published

Source code (zip) をダウンロードします。

Assets

<pre>@MPFS_ICICLE_AXI4_STREAM_DEMO_2024_09.zip</pre>	11.6 MB	Oct 11
OMPFS_ICICLE_BASE_DESIGN_2024_09.zip	11.7 MB	Oct 9
<pre> @MPFS_ICICLE_DRI_CCC_DEMO_2024_09.zip </pre>	11.6 MB	Oct 9
OMPFS_ICICLE_I2C_LOOPBACK_2024_09.zip	11.6 MB	Oct 9
OMPFS_ICICLE_MICRON_OSPI_2024_09.zip	11.6 MB	Oct 9
OMPFS_ICICLE_SPI_LOOPBACK_2024_09.zip	11.6 MB	Oct 9
Source code (zip)		Oct 9
Source code (tar.qz)		Oct 9



参考: 古い R	elease を入手する場合			
Releases のリ	ンクから、過去の Release を)	入手可能です。		
Releases 19				
S v2024.09 (La on Oct 11	atest			
+ 18 releases				
Oct 11	v2024.09 (Latest)			
◊ v2024.09 ⊘ •0- 7ab8119 Compare *	Icicle Kit Reference Design Rele	ase v2024.09		
	Changes since last release			
	Update CoreAXI4DMAController and LSRAM core versions			
	lested Libero version			
	This release has been tested with Libero SoC v2023.2			
	Pre-built programming files			
		ing a Flack Day Furgery and an array file in	huling book ship	
	The MMP32UCLE DASE_DESIGN_2024_09.20 Perives elasses 2024.09 Libero FDGA design and Hart Software Services release v2024.09 This programming file can be used to program an Icicle Kit's FPG FPGA flow or building the HSS.	ins a riashPro express programming me inc :A and eNVM without having to run throug	h the Libero	
	▼ Assets ®			
	OMPFS_ICICLE_AXI4_STREAM_DEMO_2024_09.zip	11.6 MB	Oct 11	
	OMPFS_ICICLE_BASE_DESIGN_2024_09.zip	11.7 MB	Oct 9	
	OMPFS_ICICLE_DRI_CCC_DEMO_2024_09.zip	11.6 MB	Oct 9	
	MPFS_ICICLE_I2C_LOOPBACK_2024_09.zip	11.6 MB	Oct 9	
	@MPFS_ICICLE_MICRON_QSPI_2024_09.zip	11.6 MB	Oct 9	
	OMPFS_ICICLE_SPI_LOOPBACK_2024_09.zip	11.6 MB	Oct 9	
	(§)Source code (2p)		Oct 9	
	(g) source code (usi.gz)		OCC 9	
Jul 5	v2024.06			
-O- 61c4ba9 Compare *	Icicle Kit Reference Design Rele	ase v2024.06		
	Changes since last release			
	There are no design changes in this release. The purpose of this the lcicle Kit Reference Design Flash Pro Express programming fi	release is to update the Hart Software Serv le to v2024.06.	ces included in	



参考: Libero SoC のフローを飛ばしたい場合

Pre-built programming files に記載されている通り、 MPFS_ICICLE_BASE_DESIGN_xxxx_xx.zip をダウンロードすると Icicle Kit の FPGA と eNVM へ書き込むための job ファイルが入手できます。

he MPFS_ICICLE_BASE_DESIGN_2024_09.zip release asset contains a FlashPro Expr	ess programming file including both this Libero FPGA desig	gn and Hart Software
ervices release v2024.09.		
his programming file can be used to program an Icicle Kit's FPGA and eNVM with	out having to run through the Libero FPGA flow or building) the HSS.
Assets a		
@MPFS_ICICLE_AXI4_STREAM_DEMO_2024_09.zip	11.6 MB	Oct 1
<pre>@MPFS_ICICLE_BASE_DESIGN_2024_09.zip</pre>	11.7 MB	Oct
OMPFS_ICICLE_DRI_CCC_DEMO_2024_09.zip	11.6 MB	Oct
MMPES ICICLE I2C LOOPBACK 2024 09 zin	11.6 MB	Oct
> PC > Windows (C:) > MPFS_ICICLE_BASE_D	ESIGN_2024_09	
^ ^		
名前		
MPFS_ICICLE_BASE_DESIGN_2024_09.JOD		
MDES ICICLE PASE DESIGN 2024-00 job dia	oct	

2-2. tcl スクリプトの実行

① GitHub からダウンロードした icicle-kit-reference-design-2024.09.zip を

解凍します。

日本語のフォルダ名を含まないよう、C ドライブや D ドライブ直下への展開を おすすめします。



Readme.md には tcl の実行方法、引数についての説明が記載されています。



 ② 下記 URL から hss-envm-wrapper.mpfs-icicle-kit-es.hex をダウンロードし、 ¥script_support フォルダ下へ格納します。
 <u>https://github.com/polarfire-soc/hart-software-</u>

services/releases/latest/download/hss-envm-wrapper.mpfs-icicle-kit-es.hex



備考:

Windows 環境では wget が動かないことがあるため

実行する tcl 内での hss-envm-wrapper.mpfs-icicle-kit-es.hex をダウンロード、

```
¥script support フォルダ下への格納を手動で行っています。
```



実施していない場合は次の Libero SoC での tcl 実行操作の際に下記エラーが出る場合が あります。



③ Libero SoC を起動します。



④ Project > Execute Script...をクリックします。



⑤ Script file にて MPFS_ICICLE_KIT_REFERENCE_DESIGN.tcl を選択

Arguments として

- MSS_BAREMETAL
- ∙ HSS_UPDATE
- GENERATE_PROGRAMMING_DATA

を設定して Run をクリックします。

注意: コピー&ペーストはおすすめしません。例えば改行コード等が 入ることにより正しく引数が認識されないことがあります。

🕑 Execute	Script	?	×
Script file:	C:/icicle-kit-reference-design-	/MPFS_ICICLE_KIT_REFERENCE_DESIGN	tcl
Arguments:	MSS_BAREMETAL HSS_UPDATE GEN	ERATE_PROGRAMMING_DATA	
🔽 Show sc	ript report		
Help		Run C	ancel



Libero SoC 下部の Log から確認できます。 以下の場合、引数は MSS_BAREMETAL しか適用されていません。 Log ● Messages ● Errors ● Warnings ● Info ● Info: Current Vault Location = C:\Microchip\Common\vault TCL_BEGIN: C:\icicle-kit-reference-design-2024.09\MPFS_ICICLE_KIT_REFERENCE_DESIGN.tcl Project Dath length ok. Ist MSS_BAREMETAL to 1 Creating a new project ● Info: This version of Libero supports only the enhanced constraint flow. The MPFS_ICICLE_project was created ■ Message Log U\T の場合、MSS_BAREMETAL、HSS_UPDATE、GENERATE_PROGRAMMING_DATA o 3 つの引数が適用されています。 Log ■ Message ● Errors ● Warnings ● Info
以下の場合、引数は MSS_BAREMETAL しか適用されていません。 log 『Messages @Errors AWarnings ①Info ①Info: Current Vault Location = C:\Microchip\Common\vault TCL_BEGIN: C:\icicle-kit-reference-design-2024.09\MPFS_ICICLE_KIT_REFERENCE_DESIGN.tcl Project path length ok. set MSS_BAREMETAL to 1 Creating a new project ①Info: This version of Libero supports only the enhanced constraint flow. The MPFS_ICICLE_project was created Message Log 以下の場合、MSS_BAREMETAL、HSS_UPDATE、GENERATE_PROGRAMMING_DATA の3つの引数が適用されています。 log 『Messages @Errors AWarnings ①Info
Log ■Messages ②Errors ▲Warnings ●Info ●Info: Current Vault Location = C:\Microchip\Common\vault TCL_BEGIN: C:\icicle-kit-reference-design-2024.09\MPFS_ICICLE_KIT_REFERENCE_DESIGN.tcl Project path length ok. set MSS_BAREMETAL to 1 Creating a new project ●Info: This version of Libero supports only the enhanced constraint flow. The MPFS_ICICLE_project was created Message Log 以下の場合、MSS_BAREMETAL、HSS_UPDATE、GENERATE_PROGRAMMING_DATA の 3 つの引数が適用されています。 Log ■Messages ②Errors ▲Warnings ●Info
<pre> Info: Current Vault Location = C:\Microchip\Common\vault TCL_BEGIN: C:\icicle-kit-reference-design-2024.09\MPFS_ICICLE_KIT_REFERENCE_DESIGN.tcl Project path length ok. set MSS_BAREMETAL to 1 Creating a new project Info: This version of Libero supports only the enhanced constraint flow. The MPFS ICICLE project was created Message Log UNTの場合、MSS_BAREMETAL、HSS_UPDATE、GENERATE_PROGRAMMING_DATA o 3 つの引数が適用されています。 Log Immessage Serrors Awarnings Info</pre>
Message Log 以下の場合、MSS_BAREMETAL、HSS_UPDATE、GENERATE_PROGRAMMING_DATA の3つの引数が適用されています。 Log ■Messages Serrors Awarnings ①Info
の3つの引数が適用されています。 Log ■Messages ⊗Errors AWarnings ①Info
Log
<pre>Info: Current Vault Location = C:\Microchip\Common\vault TCL_BEGIN: C:\icicle-kit-reference-design-2024.09\MPFS_ICICLE_KIT_REFERENCE_DESIGN.tcl Project path length ok. set MSS_BAREMETAL to 1 set HSS_UPDATE to 1 set GENERATE_PROGRAMMING_DATA to 1 Itreating a new project. Message Log</pre>



⑥ スクリプト実行(デザイン作成~Generate FPGA Array Data)が完了するまで しばらく待ちます。

スクリプト実行中、今何をしているかは Libero SoC 左側の Design Flow タブで 確認できます。



2-3. 書き込み







② Run PROGRAM Action をダブルクリックし、FPGA ヘデザインを書き込みます。



参考: 書き込んだデザインには既に Software が含まれています。 Tera Term 等を起動して電源を入れると動いていることを確認できます。 ここで動いているソフトウェアは GitHub からダウンロードして ¥script_support フォルダへ格納した hss-envm-wrapper.mpfs-icicle-kit-es.hex です。 Libero SoC にて Configure Design Initialization Data and Memories を開き eNVM タブを開き、BOOT_MODE_1_ENVM_CLIENT を右クリックし、Edit を選択する と確認できます。 🚳 Configure Programmer Select Programmer 🖻 🕨 🕨 Program Design • Generate FPGA Array Data L/ • Configure Design Initialization Data and Memories Generate Design Initialization Data A Configure I/O States During JTAG Programming Configure Programming Options Configure Security Design F… Design Hierar… Stimulus Hierar… Cat… Сс



Apply	Discard	He	elp					
Usage statistics			_Clients -					
Available memor	y (in pages):	512						
Used memory (in	n pages):	458	Add .		Edit	Delete		
Free memory(ir	n pages):	54						E 10
				Client	Name	Start Page	Number of bytes	End Pag
			1 BOO	T_MODE_1	_ENVM_CLIEN	ΙТ 0	117248	457
						Edit		
						Delete		
Edit DiainText Roo	t Mode 1 clier							2
Edit PlainText Boo	t Mode 1 clier	nt						?
Edit PlainText Boo	t Mode 1 clier	nt _CLIENT						?
Edit PlainText Boo ent name: BOOT_N	t Mode 1 clier	nt LOLIENT						?
Edit PlainText Boo ent name: BOOT_N NVM Content from file	t Mode 1 clier MODE_1_ENVM	nt LOLIENT	reference-de	2024.09	/script_support/	/hss-envm-wrap	per.mpfs-icicle-kit-es.h	? .:.
Edit PlainText Boo ent name: BOOT_M NVM Content from file Format:	t Mode 1 clier MODE_1_ENVM e: C:/ic	nt _CLIENT sicle=kit=r Hex	reference-de	esign-2024.08	l∕script_suppo <mark>rt∕</mark>	'hss-envm-wrap	per.mpfs-icicle-kit-es.h	?
Edit PlainText Boo ent name: BOOT_N NVM Content from file Format: Base Add	t Mode 1 clier MODE_1_ENVM a: C:/ic JInteI-H ress: ①0x202	nt _CLIENT icle-kit-r Hex 20000	reference-de	esign-2024.09	/script_suppo <mark>rt/</mark>	[/] hss-envm-wrap	per.mpfs-icicle-kit-es.h	? :
Edit PlainText Boo ent name: BOOT_N NVM Content from file Format: Base Addu itart rage (decimal)	t Mode 1 clien MODE_1_ENVM e: C:/ic Intel-P ress: ① 0x202	nt _CLIENT iicle-kit-r Hex 20000	reference-de	esign-2024.08	/script_suppo <mark>rt/</mark>	['] hss-envm-wrap	per.mpfs-icicle-kit-es.h	?
Edit PlainText Boo ent name: BOOT_M NVM Content from file Format: Base Add Start page (decimal)	t Mode 1 clien MODE_1_ENVM e: 0:/io Intel=H ress: 0 0x202 : 0	nt _CLIENT iicle-kit-r Hex _20000	reference-de	esign-2024.09	i/script_suppor <mark>t/</mark>	'hss-envm-wrap	per.mpfs-icicle-kit-es.h	? :
Edit PlainText Boo ent name: BOOT_N NVM Content from file Format: Base Addu Start page (decimal) Number of bytes (de	t Mode 1 clien MODE_1_ENVM a: C:/ic InteI-H ress: ①0x202 : 0 acimal): 102	nt _CLIENT ;icle=kit=r Hex ;20000 	reference-de	rsign-2024.09	I/script_suppo <mark>rt/</mark>	'hss-envm-wrap	per.mpfs-icicle-kit-es.h	? :
Edit PlainText Boo ent name: BOOT_N NVM Content from file Format: Base Addu Start page (decimal) Number of bytes (de	t Mode 1 clien MODE_1_ENVM e: C:/ic Intel-H ress: Ox 202 : O ecimal): 102	nt _CLIENT icle-kit-r Hex 20000	reference-de	esign-2024.08	I∕script_suppo <mark>rt∕</mark>	'hss-envm-wrap	per.mpfs-icicle-kit-es.h	?
Edit PlainText Boo ent name: BOOT_N NVM Content from file Format: Base Addu Start page (decimal) Number of bytes (de	t Mode 1 clien MODE_1_ENVM e: C:/ic Intel-H ress: Ox 202 : O ecimal): 102	nt _CLIENT icle-kit-r Hex 20000 	reference-de	esign-2024.08	I/script_suppo <mark>rt/</mark>	[/] hss-envm-wrap	per.mpfs-icicle-kit-es.h	? ex
Edit PlainText Boo ent name: BOOT_N NVM Content from file Format: Base Addi Start page (decimal) Jumber of bytes (de Include this client	t Mode 1 clien MODE_1_ENVM e: C:/ic Intel-F ress: OX_202 : O ecimal): 102	nt _CLIENT iicle-kit-r Hex 20000 	reference-de 0x0 402 tion	esign-2024.09	/script_suppor <mark>t/</mark>	'hss-envm-wrap	per.mpfs-icicle-kit-es.h	?



- 3. SoftConsole を用いたベアメタルアプリケーションの実行
- 3-1. PolarFire SoC Bare Metal Examples のダウンロード

PolarFire SoC Bare Metal Examples は GitHub からダウンロードします。 https://github.com/polarfire-soc/polarfire-soc-bare-metal-examples

最新版はページ右側のリンクから入手可能です。

Product × Solutions × Resources ×	Open Source \vee Enterprise	≥ ∨ Pricing (Q Search or jump to	Sign in Sign up
polarfire-soc / polarfire-soc-bare-me	tal-examples (Public)		Ų	Notifications
⇔ Code ⊙ Issues 5 \$\$ Pull requests 3	Actions Projects	③ Security 🗠 Insights		
🐉 main 👻 🐉 1 Branch 🛇 9 Tags		Q. Go to file	↔ Code ×	About
Charles Haden and HA-harshit power-saving: u	pdate for clock scaling examples	□ ✓ 96d8bc7 · 7 month	s ago 🕚 274 Commits	Bare metal example software projects for PolarFire SoC
🛅 .github	Add contribution docs		2 years ago	CD Readme
applications	Merged applications exampl	e code version b555cfa0f69e1	fa 10 months ago	See Code of conduct Arr Activity
driver-examples	power-saving: update for clo	ock scaling examples	7 months ago	E Custom properties
images	improve description of the b	uild configurations and debug	I 2 years ago	☆ 28 stars
README.md	Readme: supported targets a	add power saving demo	10 months ago	থ 14 forks
README Solution Code of conduct			:=	Report repository
PolarFire SoC Bare Me	tal Examples			Releases 9 2024.05 (Latest) on May 30
This repository provides bare metal embedde	d software example projects	with Microchip's SoftCons	ole IDE.	+ 8 releases

2024.05 (Latest)

n Rohitgirase released this May 30 🔊 2024.05 - • 96d8bc7

PolarFire SoC Bare metal Example Projects

- MMS MPFS HAL
- mpfs-hal-power-saving-demo project showcasing the implementation of Power Saving Modes using Clock Scaling.
 This update will demonstrate how an application can dynamically change the MSS CLK frequency to save power.
- MSS USB
- Update FatFs version from R0.09b to R0.15 in mpfs-usb-host-msc example project.
- Ethernet MAC
 Maintenance update.
- CoreOSPI

Added a sample project.

Driver	Revision
mss-mpfs-hal-example	2.4.103
mss-usb-example	2.2.107
mss-ethernet-mac-examples	1.8.120

Assets 2
Source code (zip)
Source code (tar.gz)



3-2. SoftConsole の起動

① SoftConsole を起動します。



② 任意のディレクトリを指定し Launch を押下。

sc SoftConsole v2022.2-RISC-V-747 Launcher	Х
Select a directory as workspace	
SoftConsole v2022.2-RISC-V-747 uses the workspace directory to store its preferences and development artifacts.	
Workspace: C:¥icicle-kit-reference-design- ¥SoftConsole ∨ Browse	
$\Box \underline{U}$ se this as the default and do not ask again	
<u>R</u> ecent Workspaces	
Launch Cancel	

③ Do you want to create a new workspace?にて OK をクリックします。



④ Welcome ページを閉じます。





3-3. RTC-time exmple のインポート

File > Import... をクリック

SC	SoftConsole -	SoftConsole	v2022.2-RISC-V-74	7
----	---------------	-------------	-------------------	---

Edit	Source	Refactor	Navigate	Search	Project
New Open Open	File Projects 1	from File S	ystem	Alt+Sh	nift+N >
Close Close	Editor All Edito	rs		C Ctrl+Sh	trl+W ift+W
Save Save Save Rever	As All			(Ctrl+S	Ctrl+S hift+S
Move. Renan Refres Conve	 ne sh ert Line D	elimiters To)		F2 F5 >
Print				(Ctrl+P
Impor Expor	t t				
Prope	rties			Alt+	Enter
Switch Restar Exit	n Workspa rt	ace			>
	Edit New Open Open Close Close Save Save Save Rever Move Renar Refres Conve Print Impor Expor Prope Switch Restat Exit	Edit Source New Open File Open Projects f Close Editor Close All Edito Save Save All Revert Move Rename Refresh Convert Line D Print Export Properties Switch Workspir Restart Exit	Edit Source Refactor New Open File Open File Open Projects from File Sy Close Editor Sy Close Editor Close All Editors Sy Save All Sy Sy Revert Nove Rename Refresh Convert Line Delimiters To Convert Line Delimiters To Properties Switch Workspace Restart Exit Sy	Edit Source Refactor Navigate New Open File Open Projects from File System Open Projects from File System Close Editor Close Editor Save Close All Editors Save Save As Save All Revert Save Move Rename Refresh Convert Line Delimiters To Print Export Properties Switch Workspace Restart Exit	Edit Source Refactor Navigate Search New Alt+SH Open File Open Projects from File System Colose Close Editor CC Close All Editors Ctrl+Sh Save Ctrl+Sh Save All Ctrl+Sh Revert Ctrl+Sh Move Rename Refresh Ctrl+Sh Convert Line Delimiters To Ctrl+Sh Print Q Export Alt+ Switch Workspace Alt+ Restart Exit

② Existing Projects into Workspace を選択し Next > 押下





 Select archive file を選択し、Browse...ボタンからダウンロードした polarfire-soc-bare-metal-examples-<バージョン>.zip を選択します。 インポートするプロジェクトとして、mpfs-rtc-time を選択し、 Finish を押します。

デフォルトでは全プロジェクトが選択されているので、適宜 Deselect All ボタンを ご活用ください。

sc Import			_		×
Import Projects Select a directory to sea	rch for existing Eclipse projects.				
O Select roo <u>t</u> directory:			~	B <u>r</u> ows	e
Select <u>a</u> rchive file:	C:¥polarfire-soc-bare-metal-examples-	.zip	\sim	B <u>r</u> ows	e
Projects:	1gv-flash (polarfire-soc-bare-metal-exam	nples-	^	<u>S</u> elect	All
mpfs-rtc-interrupt	(polarfire-soc-bare-metal-examples- larfire-soc-bare-metal-examples-	/driver-e /driver-exam		<u>D</u> eselec	t All
mpfs-spi-flash (po	olarfire-soc-bare-metal-examples- slave (polarfire-soc-bare-metal-examples	/driver-exan s- /driv	~	R <u>e</u> fre	sh
Options Search for nested pro Copy projects into we Close newly imported Hide projects that alr	ojects orkspace d projects upon completion ready exist in the workspace				
Working sets					
Add projec <u>t</u> to work	ing sets			Ne <u>w</u> S <u>e</u> lect	
?	< <u>B</u> ack <u>N</u> ext >	<u>F</u> inish		Cance	21



補足:
プロジェクトについての説明(How to use this example)は README.md から確認可能です。
GitHub 上からも README.md の内容を確認可能です。
https://github.com/polarfire-soc/polarfire-soc-bare-metal-examples/tree/main/driver-
examples/mss/mss-rtc/mpfs-rtc-time
Windows (C:) > icicle-kit-reference-design- > SoftConsole > mpfs-rtc-time
名前
.settings
LIM-Debug-IcicleKitEs
src src
cproject
gitignore
project
mpfs-rtc-time hw all-harts attach.launch
impfs-rtc-time hw all-harts debug.launch
mpfs-rtc-time renode all-harts debug.lau
mpfs-rtc-time renode all-harts start-platf
Market README.md

④ mpfs-rtc-time がインポートされたことを確認します。





3-4. ビルド前の編集

- ① src/boards/icicle-kit-es/platform_config/lim-debug/mpfs_hal_config/mss_sw_config.h を開きます。
 - (1) <code>#define MPFS_HAL_FIRST_HART 0 \And #define MPFS_HAL_FIRST_HART 1</code>

へ変更します。	
🔁 Project Explorer 🗙 🕞 🖻 🖻	▶ mss_sw_config.h ×
✓ 👺 mpfs-rtc-time	55 * Set MPFS_HAL_LAST_HART to a valu
> 🗊 Includes	56 * all U54_x harts.
∽ 👺 src	57 * Harts that are not started will
> 🕞 application	58 * through some other method.
x 🖻 boards	59 * The value of MPFS_HAL_FIRST_HART
v 🖨 icicle-kit-es	60 The value of MPFS_HAL_LASI_HART
	61 * A typical use-case where you set
> 🗁 rpga_design	62 * MPFS_HAL_LAST_HART = 1 1s when
✓ ➢ platform_config	63 * your application is running on U
> 🗁 ddr-release	64 * your application to the target m
> 🗁 envm-scratchpad-release	
🗸 🗁 lim-debug	67 #define MDES HAL FIRST HART
> 🗁 linker	68 #endif
✓	69
> h mss_sw_config.h	70⊖ #ifndef MPFS_HAL_LAST_HART
> 🗁 lim-release	71 #define MPFS_HAL_LAST_HART 4
> 🗁 mpfs-discovery-kit	72 #endif
> 🖻 middleware	73
> 🖨 platform	749/*
/ plationin	75 * TMAGE LOADED BV ROOTLOADER

(2) **#define** IMAGE_LOADED_BY_BOOTLOADER 0 \pounds

```
#define IMAGE_LOADED_BY_BOOTLOADER 1
```

へ変更します。

🔁 Project Explorer 🗙 🕞 😫 🍸 🕴 🗖	▶ *mss_sw_config.h ×
✓	86 * Not defined => This program assumes the
> 🔊 Includes	87 * already performed (Typ:
× 🛱 src	88 *
	89 * List of items initialised when MPFS_HA
	90 * - load virtual rom (see load_virtual_ro
V 🗁 boards	91 * - 12 cache config
🗸 🗁 icicle-kit-es	92 * - Bus error unit config
> 🗁 fpga_design	93 * - MPU config
🗸 🗁 platform_config	94 * - pmp config
> 🗁 ddr-release	95 * - I/O, clock and clock mux's, DDR and S
> 😑 envm-scratchnad-release	96 * - will start other harts, see text des
, Chim debug	97 * MPFS_HAL_LAST_HART above
V 🗁 IIII-debug	98 *
> 🔁 linker	99 */
🗸 🗁 mpfs_hal_config	100 #define IMAGE_LOADED_BY_BOOTLOADER_1
> h mss_sw_config.h	101 [©] #if (IMAGE_LOADED_BY_BOOTLOADER == 0)
> 🗁 lim-release	<pre>102 #define MPFS_HAL_HW_CONFIG</pre>
> 🗁 mpfs-discovery-kit	103 #endif
	104



② src/application/hart0/e51.cのファイル名を e51.org へ変更します。

 Project Explorer > Project Explorer ><th>< 🖻 😫 🖓 🖇 🗖 on 1.c</th><th>□</th><th> Project Explorer × Smpfs-rtc-time Includes Ssc Spape application hart0 e51.c.o hart1 hart2 </th><th>rg</th><th></th>	< 🖻 😫 🖓 🖇 🗖 on 1.c	□	 Project Explorer × Smpfs-rtc-time Includes Ssc Spape application hart0 e51.c.o hart1 hart2 	rg	
参考: tcl で生成したデザイン Subsystem (MSS)の構 元の xml ファイルは、 取ることをおすすめし	をそのまま使用せす 成を変更している場 ファイル名を xml か ます。	デ、MSS (合 xml フ ゝら xml.or	Configurator で Micr ァイルを差し替えま rg へ変更する等バッ	rocontroller ミす。 ハクアップを	
Project Explorer × Structure Project Explorer × Includes Project Explorer × Includes Project Explorer ×	In description S_Icicle_MSS_Baremetal.xn offiguration onfig	nl			



3-5. Build Configurations の選択

プロジェクトを右クリック > Build Configurations > Set Active を選択し LIM-Debug-IcicleKitEs を選択します。

눱 Project Explorer	×	E 😫 7 🕴			
 Smpfs-rtc-time Smpfs-rtc-time Smpfs-rtc-time Smpfs-rtc-time mpfs-rtc-time mpfs		New Go Into	>		
		Open in New Window Show In Show in Local Terminal	Alt+Shift+W > >		
		Copy Paste Delete Source Move Rename	Ctrl+C Ctrl+V Delete >		
		Import Export			
		Robot Framework Build Project Clean Project Refresh Close Project Close Unrelated Project	> F5		
		Build Configurations	>	Set Active	>
		Build Targets	>	Manage	

1 DDR-Release-DiscoveryKit (Execute from cached DDR memory - typically via a bootloader. Optimized (-Os).)

2 DDR-Release-IcicleKitEs (Execute from cached DDR memory – typically via a bootloader. Optimized (-Os).)

3 LIM-Debug-DiscoveryKit (Download to and debug from LIM memory. Not optimized(-00).(Could be used with boot mode 0))
4 LIM-Debug-IcicleKitEs (Download to and debug from LIM memory. Not optimized(-00).(Could be used with boot mode 0))

5 LIM-Release-DiscoveryKit (Download to and debug from LIM memory. Optimized (-Os). (Could be used with boot mode 2))

6 LIM-Release-Icicle-Kit-Es (Download to and debug from LIM memory. Optimized (-Os). (Could be used with boot mode 2))

7 eNVM-Scratchpad-Release-DiscoveryKit (Booting from eNVM, program relocates itself to scratchpad memory and continues execution. Optimized (-Os). (Could be used with boot mode 1)) 8 eNVM-Scratchpad-Release-IcicleKitEs (Booting from eNVM, program relocates itself to scratchpad memory and continues execution. Optimized (-Os). (Could be used with boot mode 1))

3-6. ビルド

プロジェクトを右クリック > Build Project を右クリックしてビルドします。







参考:
ビルドに失敗する場合、プロジェクトのインポートからやり直すとビルドできる場合があ
ります。
🛐 Problems 🔝 Markers 📮 Console 🗙 🐙 Terminal 🖋 Search 🖳 Debugger Console 🕕 Memory 🗧 🗖
🗶 4- 6- 6- 6- 6- 6- 6- 6- 6- 6- 6- 6- 6- 6-
CDT Build Console [mpfs-rtc-time]
./src/platform/hal/hw_reg_access.o ./src/platform/drivers/mss_rtc/mss_rtc.o ^
./src/platform/drivers/mss/mss_mmuart/mss_uart.o ./src/application/inc/uart_mapping.o
./src/application/hart4/u54 4.o ./src/application/hart3/u54 3.o
./src/application/hart2/u54 2.o ./src/application/hart1/u54 1.o
c:/microchip/softconsole-v2022.2-risc-v-747/riscv-unknown-elf-gcc/bin//lib/gcc/riscv64-
unknown-elf/8.3.0////riscv64-unknown-elf/bin/ld.exe:
/src/platform/mofs hal/startup gcc/mss entry.o: in function .10 ':
C:\icicle-kit-reference-design-2024 00\SoftConsole\mnfs-rtc-time\LIM-Debug-
TricleKitEs/ /src/nlatform/mfs hal/startup grc/mss entry S:300: undefined reference to
inst and
collect2 aver annor: 1d naturned 1 avit status
whole 1. *** make is a recurrent to the status
make[1]: ""[makerile://: mpis-ruc-lime.eit] Error 1
make: **** [maketile:03: all] Error 2
make all terminated with exit code 2. Build might be incomplete.
00:28:20 Build Failed. 4 errors, 0 warnings. (took 26s.28ms)



3-7. Terminal の立ち上げ

Tera Term 等任意の Terminal を立ち上げます。 複数ポートあるため、それぞれ別ウィンドウで立ち上げます。

🔟 Tera Term - [未接続]	VT							\times
ファイル(F) 編集(E) 設定	(S) コントロール(O)	ウィンドウ(W)	ヘルプ(H)					
								^
	Tera Term: シリア	ルポート 設定と	接続					×
	ポート(P)	:	COM46	~	新規>	オープン	(N)	
	スピード(E):	COM40					
	データ(D)):	COM53 COM54		牛	ャンセル	,	
	パリティ(ル	4):	none	~				
	ストップビ	、ット(S):	1 bit	\sim	~	ルプ(H)		
	フロー制行	卸(F):	none	\sim				
	ſ	送信遅延						
		0	ミリ秒/字(C)	0] ミリ秒/行	(L)		
	Device Fr Device In: Device Ma Provider N Driver Da Driver Ve	iendly Nam stance ID: I anufacturei Name: Silico te: 5–23–20 rsion: 6.7.6.	ne: Silicon Lab USB¥VID_100 r: Silicon Labe on Laboratori 018 2130	os Quad 04&PID_6 s es Inc.	CP2108 US EA71&MI_00	iB to U/)¥7&284	ART E 1493A	
	<						>	



3-8. Icicle Kit の電源投入

- ① Icicle Kit の SW4 を押し、デバイスをリセットします。
- ② 2 つの Terminal 上でメッセージが表示されます。
 (デバイスの eNVM 領域にソフトウェアも書き込まれているため。)
 COM ポート番号はお手元の環境により変わります。
 UART0 (e51) HSS(First Stage Boot Loader):



UART1 (u54_1)





 ③ UART0 (e51) - HSS(First Stage Boot Loader)が表示された Terminal にて Enter キーを押し、コマンド入力ができるようにします。
 >>

が表示され、コマンド入力できるようになったら

USBDMSC と入力します。

なお、入力可能なコマンドは HELP コマンドで確認可能です。

💆 COM46 - Tera Term VT	_		×
ファイル(E) 編集(E) 設定(S) コントロール(Q) ウィンドウ(W) ヘルプ(H)			
[0.916379] startup_service :: [init] -> [boot] [0.922489] ipi_polT_service :: [Init] -> [Monitoring] [3.79140] usbdmsc_service :: [Init] -> [Idle] Press a key to enter CLI, ESC to skip Timeout in 1 second .[3.862666] Character 13 pressed			~
[3.867249] Type HELP for list of commands [3.872596] loop 289882 took 472460191 ticks (max 472460191 ticks) >> help			
BOOT RESET HELP VERSION UPTIME DEBUG MEMTEST OSPI EMMC MMC SDCARD	PAYLOA	DSPIU	JS
BOMSC FCC			
<pre>IV USDUMBL [17.923363] usbdmsc_service :: [Idle] -> [WaitForUSBHost] [17.930524] initialize MMC [17.934439] Attempting to select SDCARD Failed [17.992595] Attempting to select eMMC Passed [18.296916] MMC - 512 byte pages, 512 byte blocks, 15273600 pages Waiting for USB Host to connect (CTRL-C to quit) </pre>	7 ticks	(may /	7
2460191 ticks)	r LICKS	(1114) ~	"
. O bytes written, O bytes read			\sim



3-9. デバッグ実行

① プロジェクトを右クリック > Debug As > Debug Configurations...を選択

Project Explorer ×	E 🕏 7 🕴 🗖		
✓		^	
> 🖑 Binaries	New	>	
> 🔊 Includes	Go Into		
🗸 👺 src	Open in New Window		
> 😥 applica	Show In	Alt+Shift+W>	
v 🗁 boards	Show in Local Terminal	>	
	Copy	Ctrl+C	
	Paste	Ctrl+V	
× 🛃 🗙	Delete	Delete	
	Source	>	
5 6	Move		
	Rename	F2	
> 🥃 🚵	Import		
> 👩 🗠	Export		
> [🕞	Robot Framework	>	
> 🗁 f	Build Project		
> 🗁 mpt	Clean Project		
> 🗁 middle	Refresh	F5	
> 🗁 plation	Close Project		
mpfs-rtc-t	Close Unrelated Project		
mpfs-rtc-t	Build Configurations	>	
mpfs-rtc-t	Build Targets	>	
mpfs-rtc-t	Index	>	
<	Profiling Tools	>	
🎄 Debug 🗙 💊 Br 🗕 🔘	Run As	>	
🕞 🔆 (🍄	Debug As	>	I Robot Test
	Profile As	>	2 Robot Remote Test
	Restore from Local History		G 3 Local C/C++ Application
	cppcheck	>	Debug Configurations



② mpfs-rtc-time hw all-harts debug をクリックします。

Startup タブを開き、Initial Reset と Set breakpoint at のチェックを外し、 Apply を押し Debug を押下します。

sc Debug Configurations	- D X
Create, manage, and run configurations	- Alton
Image: Second	Name: mpfs-rtc-time hw all-harts debug Main Main Store Source Common Source SVD Path Initialization Commands Initialization Commands Initialization Commands Initialization Commands Initial Reset Type: init Enable Arm semihosting Load Symbols and Executable Load Symbols Use project binary: mpfs-rtc-time.elf Use file: Workspace File System Symbols offset (hex): Load executable Use project binary: mpfs-rtc-time.elf Use project binary: mpfs-rtc-time.elf Use project binary: mpfs-rtc-time.elf Use project binary: mpfs-rtc-time.elf Use file: Workspace File System Executable offset (hex): Runtime Options Debug in RAM Run/Restart Commands Pre-run/Restart reset Type: (always executed at Restart) thread apply all set \$pc=_start Set program counter at (hex): Set breakpoint at: e51 Continue
Filter matched 6 of 13 items	Re <u>v</u> ert Apply
?	Debug Close

備考:

今回の演習では一例として Debug 設定を変更して実行しましたが、特にこの設定にすべき 等の決まりはありません。

例えば、Initial Reset.のチェックを外した場合、ターゲットリセットはスキップされ、現在のCPUの状態からアプリケーションが実行されてしまいます。

デフォルト設定の状態で実行する、設定を変更する等 任意に使用下さい。



デバッガについては GitHub 上に記載がありますので併せてご参考ください。 引用「You may change the existing debug launchers or create your own launcher to suit your project.」 <u>https://github.com/polarfire-soc/polarfire-soc-bare-metal-examples?tab=readme-ov-</u>

③ Terminal 上に1秒ごとに経過時間が表示されることを確認します。 ソースコードは u54_1.c をご参考ください。

```
snprintf((char *)display_buffer, sizeof(display_buffer),
```

```
"Seconds: %02d",(int)(calendar_count.second));
```

ファイル(E) 編集(E) 設定(S) コントロール(Q) ウィンドウ(W) ヘルプ(H) Seconds: 37 Seconds: 38 Seconds: 39 Seconds: 40 Seconds: 41	7 - Tera Term VT – 🗆 🗙
Seconds: 37 Seconds: 38 Seconds: 39 Seconds: 40 Seconds: 41	編集(E) 設定(S) コントロール(Q) ウィンドウ(W) ヘルプ(H)
Seconds: 42 Seconds: 43 Seconds: 44 Seconds: 45 Seconds: 46 Seconds: 47 Seconds: 47 Seconds: 48 Seconds: 49 Seconds: 50 Seconds: 50 Seconds: 51 Seconds: 52 Seconds: 53 Seconds: 53 Seconds: 55 Seconds: 56	37 38 39 40 41 42 43 44 45 46 46 47 48 49 50 51 52 53 54 55 56 56

以上



file#debug-launchers

変更履歴

リビジョン	日付	概要
V1	2024年1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。
- 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。

