# 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	
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	
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	27



#### 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. Icicle 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

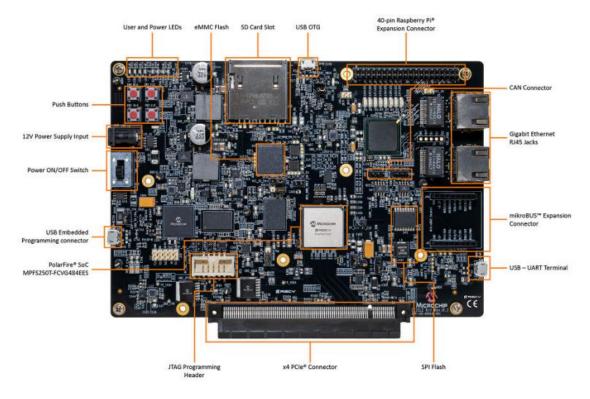
#### 1-2. Silicon Labs CP2108 drivers のインストール

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
<a href="https://ww1.microchip.com/downloads/aemDocuments/documents/FPGA/ProductDocuments/UserGuides/microchip\_polarfire\_soc\_fpga\_icicle\_kit\_user\_guide\_vb.pdf#page=14">https://ww1.microchip.com/downloads/aemDocuments/documents/FPGA/ProductDocuments/UserGuides/microchip\_polarfire\_soc\_fpga\_icicle\_kit\_user\_guide\_vb.pdf#page=14</a>

#### 1-3. Icicle 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

 $C: \verb§§Microchip§§ SoftConsole-v2022.2-RISC-V-747§ softconsole.cmd$ 

ファイル内で環境変数を設定。

```
set FPGENPROG=C:\Microchip\Libero_SoC_v2024.2\Designer\bin64\fpgenprog.exe

set JDCGEN=C:\Microchip\Libero_SoC_v2024.2\Designer\bin\jdcgen.exe

REM Setting our PATHS

TO DESIGN OF THE PARTY PA
```

# 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。

```
REM ----- PolarFireSoC boot mode programming ------
REM For PolarFireSoC boot mode programming using mpfsBootmodeProgrammer.jar the fpgenprog
REM tool must be installed. Fpgenprog is shipped with 'Libero' and with 'Program and Debug'
       REM tools and SoftConsole must know where it is installed.
       REM The mpfsBootmodeProgrammer.jar will try to autodetect the fpgenprog tools if it's REM installed in the default locations, more details about that here:
        {\tt REM \ https://mi-v-ecosystem.github.io/SoftConsole-Documentation/SoftConsole-v2022.2/using \ softconsole/mpfs.html}
       REM
        REM However, if fpgenprog path can't be autodetected then it must be set by the user.
       REM Either by setting it in SoftConsole External Tool launch configuration's Environment REM tab for each launcher. Or once globally by setting it here by:
13
14
15
16
17
18
19
        REM set FPGENPROG=<FULL_ABSOLUTE_PATH_TO_FPGENPROG_BINARY>
        REM Example (full path including the binary filename):
       REM set FPGENPROG=C:\Microsemi\Libero_SoC_v2022.2\Designer\bin64\fpgenprog.exe
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
        REM
                                         - JDC Helper -
       REM For jdchelper.jar the jdcgen tool must be installed. Jdcgen is shipped with:
REM 'Libero' and with 'Program and Debug' tools and SoftConsole must know where it is installed.
       REM
       The jdcHelper.jar will try to autodetect the jdcgen tools if it's

REM installed in the default locations, more details about that here:

REM https://mi-v-ecosystem.github.io/SoftConsole-Documentation/SoftConsole-v2022.2/using softconsole/mpfs.html
       REM
       REM However, if jdcgen path can't be autodetected then it must be set by the user.
       REM Either by setting it in SoftConsole External Tool launch configuration's Environment tab for each launcher. Or once globally by setting it here by:

REM Set JDCGEN=<FULL_ABSOLUTE_PATH_TO_FPGENPROG_BINARY>
       REM
       REM Example (full path including the binary filename):
        REM
               set JDCGEN=C:\Microsemi\Libero_SoC_v2022.2\Designer\bin\jdcgen.exe
```

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



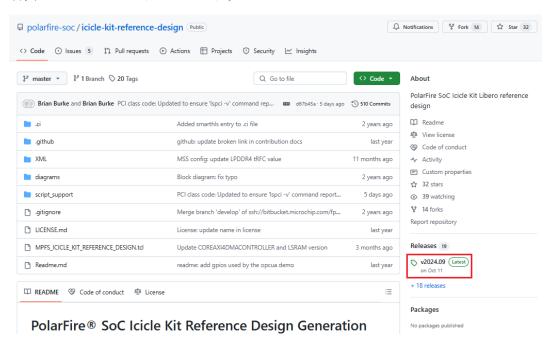
### 2. Libero SoC プロジェクトの用意

### 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

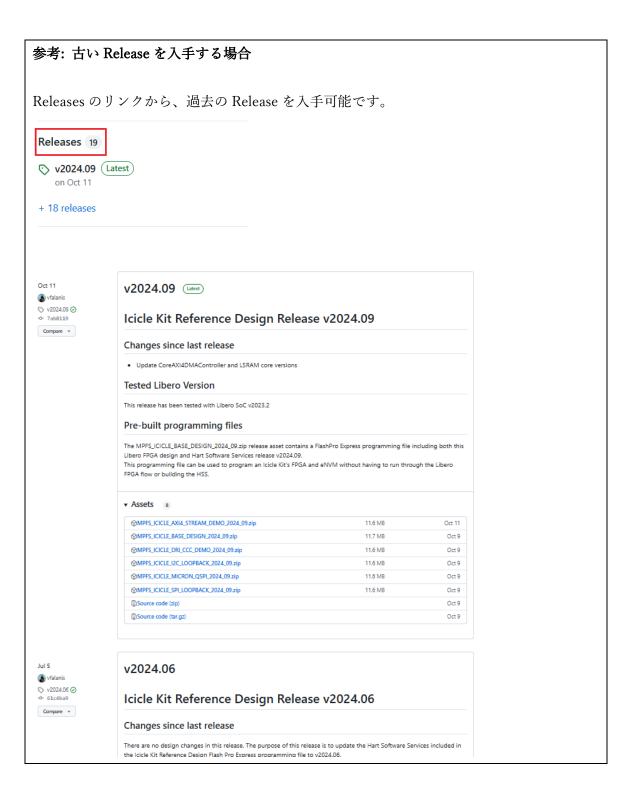
ここでは例として現在の最新版 v2024.09 を使用します。 右側の v2024.09 をクリックします。



Source code (zip) をダウンロードします。

▼ Assets 8		
	11.6 MB	Oct 11
	11.7 MB	Oct 9
₱MPFS_ICICLE_DRI_CCC_DEMO_2024_09.zip	11.6 MB	Oct 9
	11.6 MB	Oct 9
	11.6 MB	Oct 9
	11.6 MB	Oct 9
🖟 Source code (zip)		Oct 9
Source code (tar.qz)		Oct 9





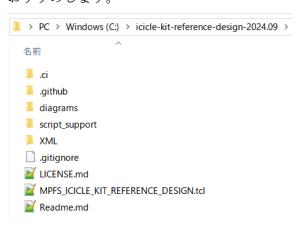




#### 2-2. tcl スクリプトの実行

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。



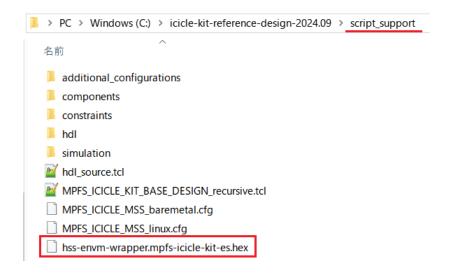
Readme.md には tcl の実行方法、引数についての説明が記載されています。



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。

https://github.com/polarfire-soc/hart-software-

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。

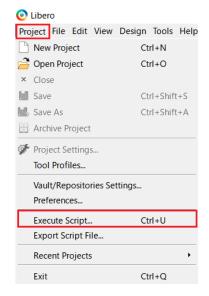
実施していない場合は次の Libero SoC での tcl 実行操作の際に下記エラーが出る場合があります。



③ Libero SoC を起動します。



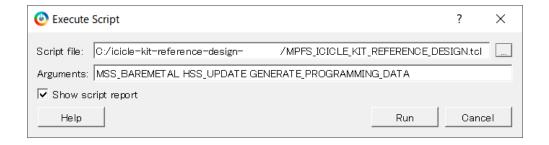
④ Project > Execute Script...をクリックします。



- ⑤ Script file にて MPFS\_ICICLE\_KIT\_REFERENCE\_DESIGN.tcl を選択 Arguments として
  - MSS\_BAREMETAL
  - HSS\_UPDATE
  - $\cdot \ \mathsf{GENERATE\_PROGRAMMING\_DATA}$

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。





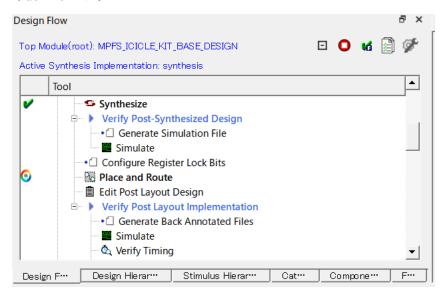
# 参考: 実行した tcl 引数が反映されているかどうかの確認 Libero SoC 下部の Log から確認できます。 以下の場合、引数は MSS BAREMETAL しか適用されていません。 1 Info: Current Vault Location = C:\Microchip\Common\vault TCL BEGIN: C:\icicle-kit-reference-design-2024.09\MPFS ICICLE KIT REFERENCE DESIGN.tcl Project path length ok set MSS BAREMETAL to 1 Creating a new project $oldsymbol{0}$ 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。 ■ Messages SErrors AWarnings Olinfo finfo: 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



Message Log

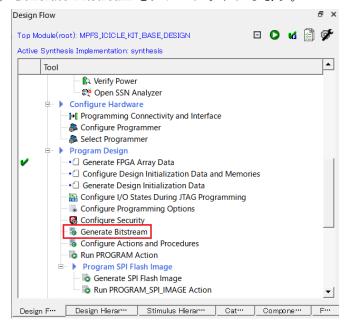
⑥ スクリプト実行(デザイン作成~Generate FPGA Array Data)が完了するまで しばらく待ちます。

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。



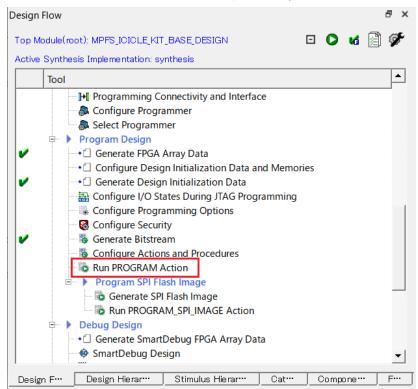
#### 2-3. 書き込み

① Generate Bitstream をダブルクリックします。





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:

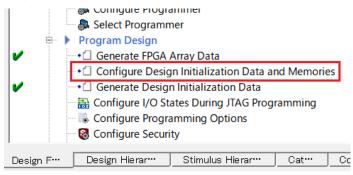
書き込んだデザインには既に Software が含まれています。

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。

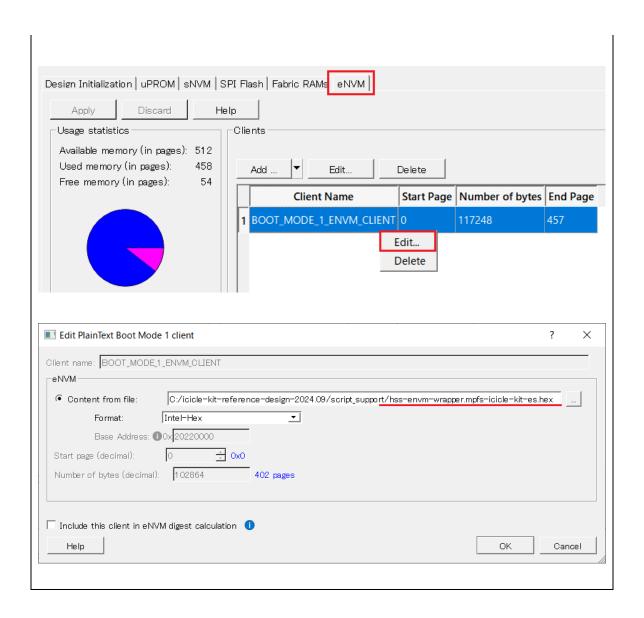
ここで動いているソフトウェアは GitHub からダウンロードして

¥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。





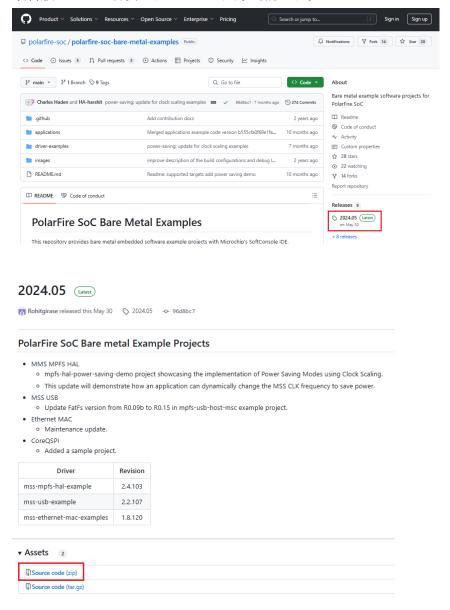




- 3. SoftConsole を用いたベアメタルアプリケーションの実行
- 3-1. PolarFire SoC Bare Metal Examples のダウンロード

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。



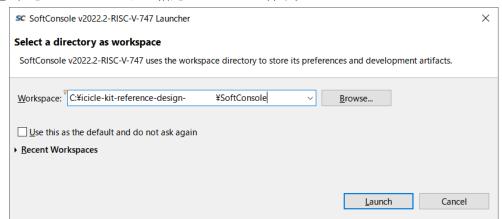


#### 3-2. SoftConsole の起動

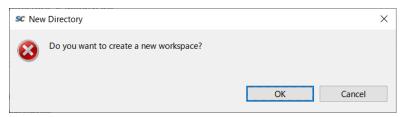
① SoftConsole を起動します。



② 任意のディレクトリを指定し Launch を押下。



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



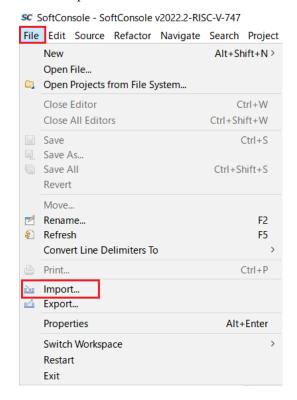
④ Welcome ページを閉じます。



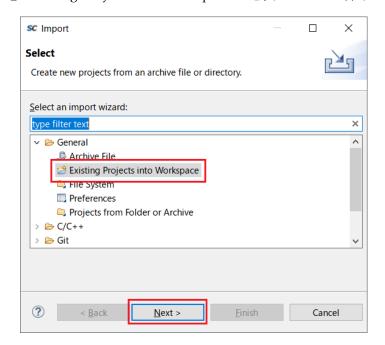


# 3-3. RTC-time exmple のインポート

① File > Import... をクリック



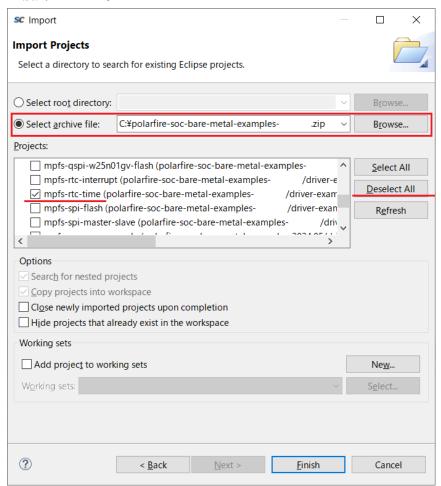
② Existing Projects into Workspace を選択し Next > 押下





③ 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。

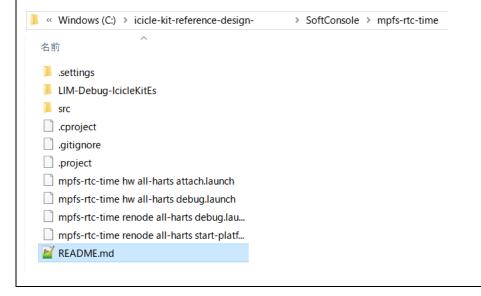




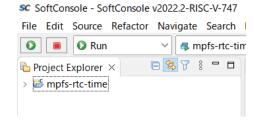
#### 補足:

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



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) #define MPFS\_HAL\_FIRST\_HART 0 を #define MPFS\_HAL\_FIRST\_HART 1

```
へ変更します。
                                                                           E S 7 8 □ □ I mss_sw_config.h ×
  Project Explorer ×
                                                                                                                                        * Set MPFS_HAL_LAST_HART to a valu
  * all U54_x harts.
                                                                                                                                        56
          > 🗊 Includes
                                                                                                                                                    * Harts that are not started will

✓ 

## src

                                                                                                                                        * through some other method.
                  > 🗁 application
                                                                                                                                        59
                                                                                                                                                    * The value of MPFS_HAL_FIRST_HART
                  v 🗁 boards
                                                                                                                                                     * The value of MPFS_HAL_LAST_HART
                                                                                                                                        60
                         61
                                                                                                                                                    * A typical use-case where you set
                                                                                                                                                    * MPFS_HAL_LAST_HART = 1 is when
                                 > 🗁 fpga_design
                                                                                                                                       62
                                                                                                                                       63
                                                                                                                                                    * your application is running on U
                                 platform_config
                                                                                                                                       64
                                                                                                                                                     * your application to the target m
                                        > 🗁 ddr-release
                                                                                                                                      65 */
                                        >  envm-scratchpad-release
                                                                                                                                      660 #ifndef MPFS HAL FIRST HART
                                        lim-debug
                                                                                                                                       67 #define MPFS_HAL_FIRST_HART
                                               > 🗁 linker
                                                                                                                                       68 #endif

w by mpfs_hal_config

mathematical config

math
                                                                                                                                       69
                                                                                                                                       700 #ifndef MPFS HAL LAST HART
                                                       > h mss_sw_config.h
                                                                                                                                       71 #define MPFS_HAL_LAST_HART
                                        > 🗁 lim-release
                                                                                                                                        72 #endif
                          > 🗁 mpfs-discovery-kit
                                                                                                                                        73
                  > 🗁 middleware
                                                                                                                                        749 /*
                  > 🗁 platform
                                                                                                                                                     * TMAGE LOADED BY BOOTLOADER
```

(2) #define IMAGE\_LOADED\_BY\_BOOTLOADER 0 を #define IMAGE\_LOADED\_BY\_BOOTLOADER 1 へ変更します。

```
Project Explorer X
                                                                                                                                            h *mss_sw_config.h ×

√ № mpfs-rtc-time

                                                                                                                                                   86
                                                                                                                                                               * Not defined
                                                                                                                                                                                                                      => This program assumes tl
                                                                                                                                                   87
                                                                                                                                                                                                                                   already performed (Typ:
         > 🛍 Includes
                                                                                                                                                   88 *
         y 🕮 src
                                                                                                                                                              * List of items initialised when MPFS_HAI
                                                                                                                                                   89
                 > 휻 application
                                                                                                                                                   90
                                                                                                                                                              * - load virtual rom (see load_virtual_ro
                 boards
                                                                                                                                                   91
                                                                                                                                                                * - 12 cache config

✓ 

icicle-kit-es

                                                                                                                                                               * - Bus error unit config
                                                                                                                                                  92
                                 > 🗁 fpga_design
                                                                                                                                                  93
                                                                                                                                                                * - MPU config
                                                                                                                                                                * - pmp config
                                                                                                                                                  94

→ 

platform_config

                                                                                                                                                               * - I/O, clock and clock mux's, DDR and S
                                                                                                                                                  95
                                          > 🗁 ddr-release
                                                                                                                                                  96 * - will start other harts, see text desc
                                          > 🗁 envm-scratchpad-release
                                                                                                                                                  97
                                                                                                                                                                               MPFS_HAL_LAST_HART above
                                          v 🗁 lim-debug
                                                                                                                                                  98
                                                  > 🗁 linker
                                                                                                                                                  99

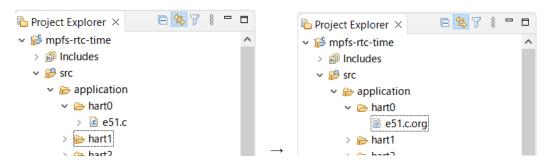
w 

mpfs_hal_config

mpfs_hal_config
                                                                                                                                               100 #define IMAGE LOADED BY BOOTLOADER 1
                                                          > In mss_sw_config.h
                                                                                                                                               1010 #if (IMAGE_LOADED_BY_BOOTLOADER == 0)
                                                                                                                                              102 #define MPFS_HAL_HW_CONFIG
                                          > 🗁 lim-release
                                                                                                                                               103 #endif
                          > 🗁 mpfs-discovery-kit
                                                                                                                                             104
```



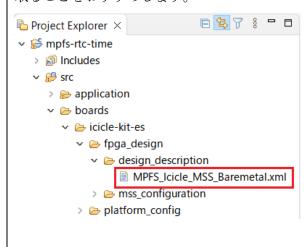
② src/application/hart0/e51.c のファイル名を e51.org へ変更します。



#### 参考:

tcl で生成したデザインをそのまま使用せず、MSS Configurator で Microcontroller Subsystem (MSS)の構成を変更している場合 xml ファイルを差し替えます。

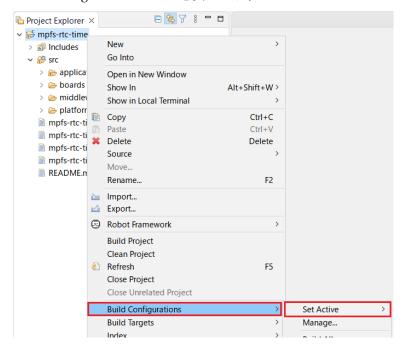
元の xml ファイルは、ファイル名を xml から xml.org へ変更する等バックアップを 取ることをおすすめします。

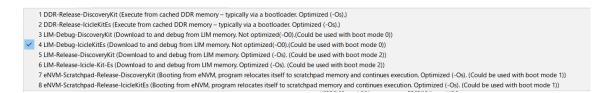




#### 3-5. Build Configurations の選択

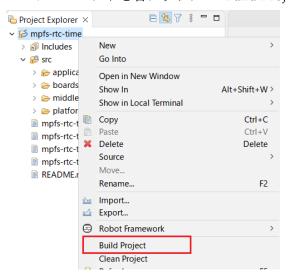
プロジェクトを右クリック > Build Configurations > Set Active を選択し LIM-Debug-IcicleKitEs を選択します。





#### 3-6. ビルド

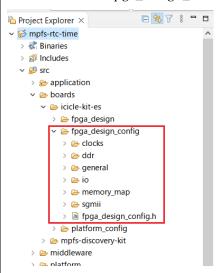
プロジェクトを右クリック > Build Project を右クリックしてビルドします。





#### 参考:

ビルドを行うと fpga\_design\_config フォルダが生成されます。



MSS Configurator で Microcontroller Subsystem (MSS)の構成を変更し、xml ファイルを 差し替えてビルドする場合には事前に"src/boards/fpga\_design\_config"配下のすべての ファイルを削除してください。

#### 参考:

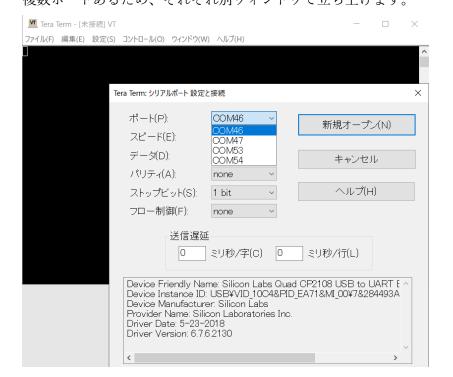
ビルドに失敗する場合、プロジェクトのインポートからやり直すとビルドできる場合があります。

```
📳 Problems 🖺 Markers 💂 Console 🗡 🧬 Terminal 🔗 Search 🖳 Debugger Console 🕕 Memory
                                                                      X | ⊕ ⊕ ⊕ | ∰ ⊕ | ₩ ₽ ▼ → →
CDT Build Console [mpfs-rtc-time]
./src/platform/hal/hw_reg_access.o ./src/platform/drivers/mss_mts_rtc/mss_rtc.o
./src/platform/drivers/mss/mss\_mmuart/mss\_uart.o \quad ./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/mpfs_hal/startup_gcc/mss_entry.o: in function `.L0 ':
C:\icicle-kit-reference-design-2024.09\SoftConsole\mpfs-rtc-time\LIM-Debug-
IcicleKitEs/../src/platform/mpfs_hal/startup_gcc/mss_entry.S:390: undefined reference to
 `main_first_hart_app'
collect2.exe: error: ld returned 1 exit status
make[1]: *** [makefile:72: mpfs-rtc-time.elf] Error 1
make: *** [makefile:63: 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。





#### 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)

```
MCOM47-Tera Term VT

ファイル色 編集(E) 設定(S) コントロール(O) ウィンドウ(M) ヘルブ(H)

DDR training ... Passed ( 2953 ms)

3.47788] DDR-Lo size is 32 MiB

[3.52371] DDR-Hi size is 1888 MiB

OpenSBI v1.2

Platform Name : Microchip PolarFire(R) SoC

Platform Features : medeleg

Platform HART Count : 5

Platform IPI Device : aclint-mswi

Platform Timer Device : aclint-mswi

Platform HSM Device : mmofs hsm

Platform Rboot Device : mofs reset

Platform Rboot Device : mofs reset

Platform Rbout Device : mofs reset

Platform Rbout Device : mofs reset
```



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-TeraTerm VT
ファイルD 編集(E) 設定(S) コットロール(O) ウィンドウ(W) ヘルブ(H)

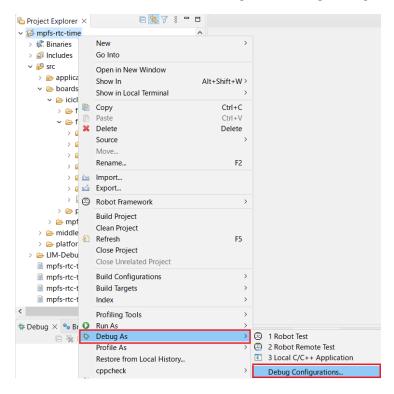
: [0.916379] startup service :: [init] -> [boot]
[0.922489] ipi polT service :: [Init] -> [Monitoring]
[3.79140] usbdmsc service :: [Init] -> [Idle]
Press a key to enter CLI, ESC to skip
Timeout in 1 second
. [3.862666] Character 13 pressed

[3.867249] Type HELP for list of commands
[3.872593] loop 289882 took 472460191 ticks (max 472460191 ticks)
>> help
Supported commands:
BOOT RESET HELP VERSION UPTIME DEBUG MEMTEST OSPI EMMC MMC SDCARD PAYLOAD SPI US RDMSC. ECC.
>> USBDMSC
[17.923603] usbdmsc service :: [Idle] -> [WaitForUSBHost]
[17.992505] Attempting to select SDCARD ... Failed
[17.992505] 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)
... 0 bytes written, 0 bytes read
```



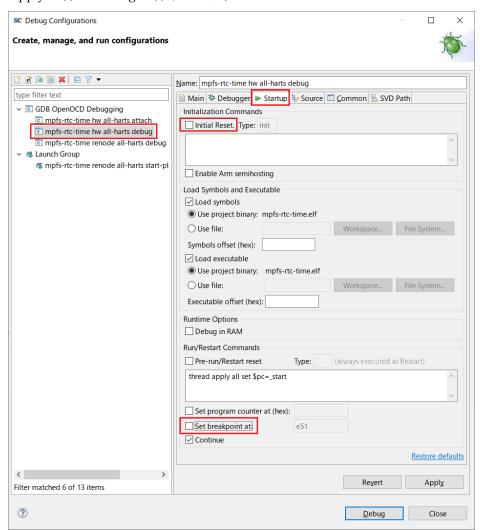
# 3-9. デバッグ実行

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





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。



#### 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.」

 $\underline{ https://github.com/polar fire-soc/polar fire-soc-bare-metal-examples? tab=readme-ov-file \# debug-launchers$ 

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。

以上



#### 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。
- 3. 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/

- 4. 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。

