



Zephyr RTOSでMatterを試してみた ～ Matter公式サンプルにないボードでも動くか～

← NAGOYA

Zephyr Project meetup: Nagoya, Japan

March 27 2026

Junichi Hirata

アジェンダ

- 自己紹介
- 今回やること
- Matterとは
- ZephyrとMatter
- ボードへのMatterをポーティングの様様 (もちろんOSはZephyr)

今回やること

- ・ZephyrでMatterを動かしたい

Matter



MATTERとは

- ・CSA (Connectivity Standards Alliance)が策定したスマートホーム向けの通信プロトコル
- ・マルチベンダー
異なるメーカーの機器でも同じプロトコルで制御できる
- ・高いセキュリティ

- ・最近のプロトコルで急速に普及している

MATTERでできることの一つ

家電の制御 (例 スマートスピーカーからの照明制御)



MATTER いろいろな機器をMATTERで

Matter SDKのexampleから


```
~/matter_zephyr/connectedhomeip/examples$ ls
BUILD.gn                               darwin-framework-tool          lit-icd-app                     rvc-app
air-purifier-app                       dishwasher-app                 lock-app                        shell
air-quality-sensor-app                energy-gateway-app           log-source-app                 smoke-co-alarm-app
all-clusters-app                      energy-management            microwave-oven-app            temperature-measurement-app
all-clusters-minimal-app              evse-app                    minimal-mdns                   terms-and-conditions-app
all-devices-app                       fabric-admin                  network-manager-app           thermostat
android                               fabric-bridge-app            ota-provider-app              thread-br-app
bridge-app                            fabric-sync                   ota-requestor-app             tv-app
build_overrides                       java-matter-controller       persistent-storage            tv-casting-app
camera-app                             jf-admin-app                 pigweed-app                   virtual-device-app
camera-controller                     jf-control-app               placeholder                    water-heater-app
chef                                   kotlin-matter-controller     platform                       water-leak-detector-app
chip-tool                              laundry-washer-app           providers                       window-app
closure-app                           light-switch-app             pump-app                       pump-controller-app
common                                lighting-app                  refrigerator-app
contact-sensor-app                   lighting-app-data-mode-no-unique-id
```

MATTER SDK

githubでconnectedhomeipが公開されています。

<https://github.com/project-chip/connectedhomeip/>

Documentation links

-  [Ask DeepWiki](#)
- [Matter SDK documentation page](#)
- [Matter SDK Coverage Report](#)

Matter

 Builds failing

Builds

MATTER SDKの中のZEPHYR

あります。connectedhomeipのexample以下をfind

(./lighting-app/zephyr_boardsは私が今回追加)

これ以外もnordicがzephyr (次のページ)

```
~/matter_zephyr/connectedhomeip/examples$ find . -name zephyr -maxdepth 4
find: warning: you have specified the global option -maxdepth after the argument -name, but global
ional, i.e., -maxdepth affects tests specified before it as well as those specified after it. Plea
ions before other arguments.
./lighting-app/silabs/zephyr
./lighting-app/realtek/zephyr
./lighting-app/zephyr_boards/build_frm/zephyr
./lighting-app/zephyr_boards/renesas/zephyr
./lighting-app/zephyr_boards/build/zephyr
./platform/nxp/zephyr
./platform/silabs/zephyr
./platform/realtek/zephyr
./all-clusters-app/nxp/zephyr
./thermostat/nxp/zephyr
./laundry-washer-app/nxp/zephyr
```

Zephyr_boardsは追加したディレ
クトリで元のサンプルにはない

MATTER SDKの中のZEPHYR

```
~/matter_zephyr/connectedhomeip/examples$ find . -name nrfconnect -maxdepth 4
find: warning: you have specified the global option -maxdepth after the argument -name, but global options are not positional, i.e., -maxdepth affects tests specified before it as well as those specified after it. Please specify global options before other arguments.
./lighting-app/nrfconnect
./platform/nrfconnect
./all-clusters-app/nrfconnect
./all-clusters-minimal-app/nrfconnect
./pump-app/nrfconnect
./lock-app/nrfconnect
./window-app/nrfconnect
./common/pigweed/nrfconnect
./lit-icd-app/nrfconnect
./shell/nrfconnect
./pump-controller-app/nrfconnect
./chef/nrfconnect
./light-switch-app/nrfconnect
```

今回やること、そのモチベーション

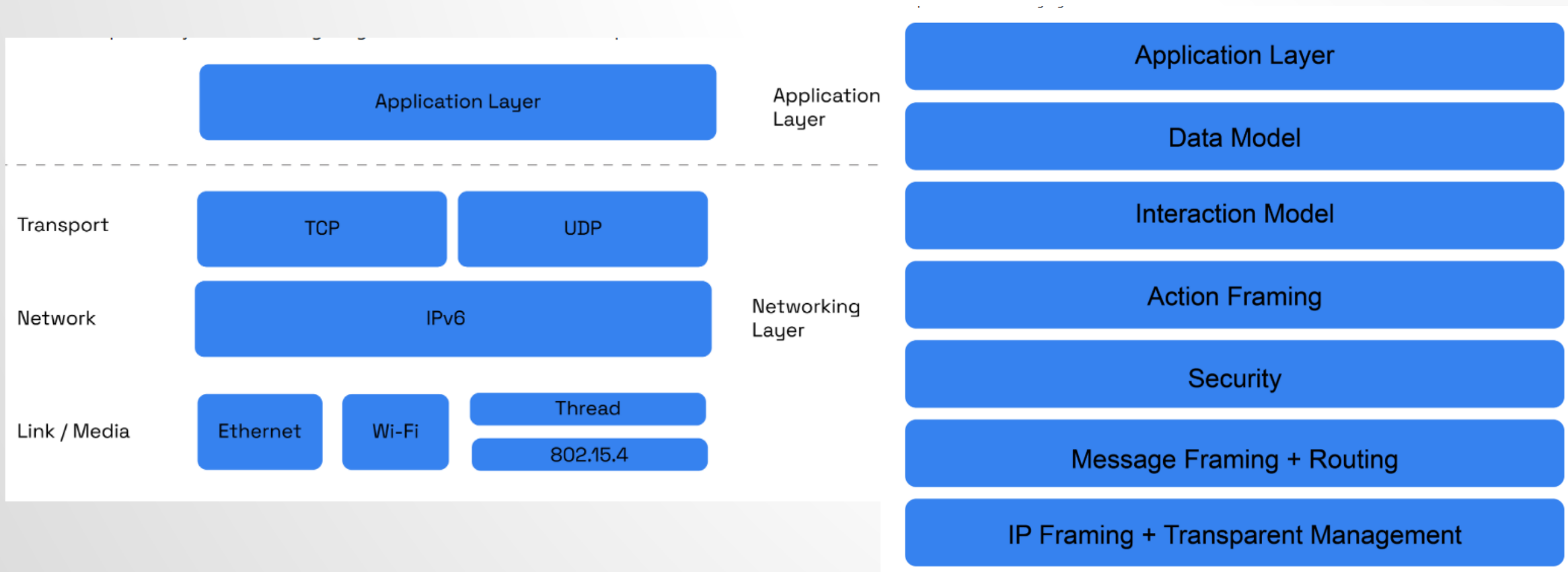
↓ 以外のメーカーのSoCを採用したボードでMatterが動くか
確認してみた。ポーティングし易いかな。

趣味の活動なので機能最小限で、実用考えなくて、楽しんでやればいい。

```
~/matter_zephyr/connectedhomeip/examples$ find . -name zephyr -maxdepth 4
find: warning: you have specified the global option -maxdepth after the argument -name, but global
ional, i.e., -maxdepth affects tests specified before it as well as those specified after it. Plea
ions before other arguments.
./lighting-app/silabs/zephyr
./lighting-app/realtek/zephyr
./lighting-app/zephyr_boards/build_frdm/zephyr
./lighting-app/zephyr_boards/renesas/zephyr
./lighting-app/zephyr_boards/build/zephyr
./platform/nxp/zephyr
./platform/silabs/zephyr
./platform/realtek/zephyr
./all-clusters-app/nxp/zephyr
./thermostat/nxp/zephyr
./laundry-washer-app/nxp/zephyr
```

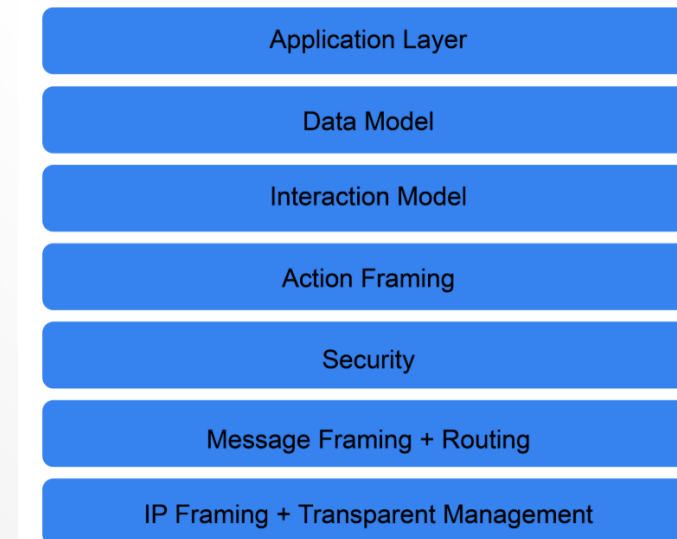
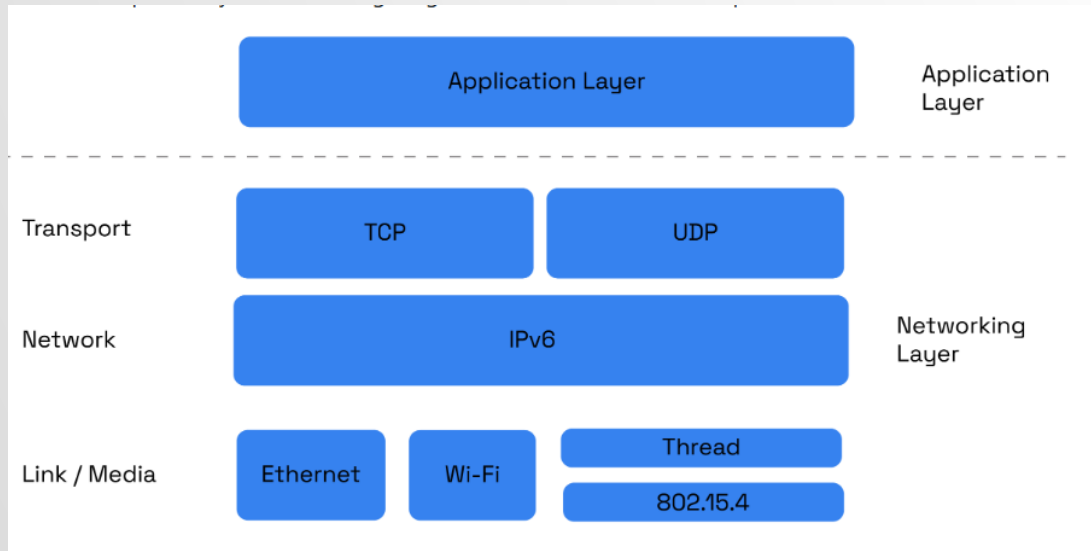
MATTERの構成

出展 <https://github.com/project-chip/connectedhomeip/>



作業前に見通し？

ZEPHYRはRTOS, TCP/IP, セキュリティ載っている、nxp, Nordic, silabs, realtekの作例あるし、やれるだろう と思ってました。



今回のターゲットボード

Nucleo H563ZI

もちろんZephyrに対応しているボードでETHERNETが動作する

Zephyrのボード説明ページ

https://docs.zephyrproject.org/latest/boards/st/nucleo_h753zi/doc/index.html

Nucleo H753ZI

Overview

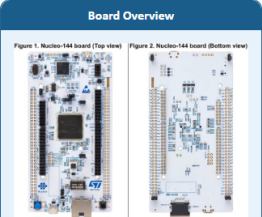
The STM32 Nucleo-144 boards offer combinations of performance and power that provide an affordable and flexible way for users to build prototypes and try out new concepts. For compatible boards, the SMPS (Switched-Mode Power Supply) significantly reduces power consumption in Run mode.

The Arduino-compatible ST Zio connector expands functionality of the Nucleo open development platform, with a wide choice of specialized Arduino* Uno V3 shields.

The STM32 Nucleo-144 board does not require any separate probe as it integrates the ST-LINK/V2-1 debugger/programmer.

The STM32 Nucleo-144 board comes with the STM32 comprehensive free software libraries and examples available with the STM32Cube MCU Package.

Key Features



Board Overview

Figure 1. Nucleo-144 board (Top view) Figure 2. Nucleo-144 board (Bottom view)

Nucleo H753ZI

Name:	nucleo_h753zi
Vendor:	STMicroelectronics
Status:	Maintained
Architecture:	arm
SoC:	stm32h753xx

[Browse board sources](#)

Hardware

Nucleo H753ZI provides the following hardware components:

- STM32H753ZI in LQFP144 package
- ARM 32-bit Cortex-M7 CPU with FPU
- Chrom-ART Accelerator
- Hardware JPEG Codec
- 480 MHz max CPU frequency
- VDD from 1.62 V to 3.6 V
- 2 MB Flash
- 1 MB SRAM

ビルド・デバッグ環境


WSL上のUbuntu24

ターゲットアプリ

lighting-app

照明制御のアプリケーション

```
~/matter_zephyr/connectedhomeip/examples$ ls
BUILD.gn          darwin-framework-tool      lit-icd-app         rvc-app
air-purifier-app  dishwasher-app             lock-app            shell
air-quality-sensor-app  energy-gateway-app        log-source-app     smoke-co-alarm-app
all-clusters-app    energy-management         microwave-oven-app  temperature-measurement-app
all-clusters-minimal-app  evse-app                  minimal-mdns        terms-and-conditions-app
all-devices-app     fabric-admin               network-manager-app  thermostat
android            fabric-bridge-app          ota-provider-app    thread-br-app
bridge-app          fabric-sync                 ota-requestor-app   tv-app
build_overrides    java-matter-controller    persistent-storage  tv-casting-app
camera-app          jf-admin-app               pigweed-app         virtual-device-app
camera-controller  jf-control-app             placeholder          water-heater-app
chef               kotlin-matter-controller   platform             water-leak-detector-app
chip-tool          laundry-washer-app         providers            window-app
closure-app        light-switch-app           pump-app
common             lighting-app                pump-controller-app
contact-sensor-app lighting-app-data-mode-no-unique-id  refrigerator-app
```



というわけで、
ポーティング作業開始

まずはざっくり、MATTR SDKのZEPHYRに 関連する個所を把握

connectedhomeip/config/zephyr ビルドに使うファイルたち

```
~/matter_zephyr/connectedhomeip/config/zephyr$ ls  
Kconfig README.md app chip-gn chip-module ota-image.cmake ota-image_sysbuild.cmake zephyr-util.cmake  
~/matter_zephyr/connectedhomeip/config/zephyr$
```

まずはざっくり、MATTR SDKのZEPHYRに 関連する個所を把握

exampleの下のzephyrを使っているメーカーのサンプル

```
~/matter_zephyr/connectedhomeip/examples$ find . -name zephyr -maxdepth 4
find: warning: you have specified the global option -maxdepth after the argument -name, but global
ional, i.e., -maxdepth affects tests specified before it as well as those specified after it. Plea
ions before other arguments.
./lighting-app/silabs/zephyr
./lighting-app/realtek/zephyr
./lighting-app/zephyr_boards/build_frdm/zephyr
./lighting-app/zephyr_boards/renesas/zephyr
./lighting-app/zephyr_boards/build/zephyr
./platform/nxp/zephyr
./platform/silabs/zephyr
./platform/realtek/zephyr
./all-clusters-app/nxp/zephyr
./thermostat/nxp/zephyr
./laundry-washer-app/nxp/zephyr
```

これらを参考、マネしよう (+ nrfconnect = NORDIC も)

ポーティング ～ ビルド成功までの道のり ～

- zephyr_boardsディレクトリを作成

```
~/matter_zephyr/connectedhomeip/examples/lighting-app$ ls
ameba  bouffalolab  infineon      linux      python  silabs  ti
asr    esp32         lighting-common  nrfconnect  qpg     stm32   tizen
beken  genio         lighting-common.back  nxp         realtek  telink  zephyr_boards
```

- CMakeList.txt, Kconfig, prj.conf、src/を作成

```
~/matter_zephyr/connectedhomeip/examples/lighting-app/zephyr_boards$ ls
CMakeLists.txt  Kconfig  boards  build  build_frdm  prj.conf  prj.conf.back_new_release  renesas  src  third_party
```

- src/の下にinclude/とC++のソースコードを作成

```
~/matter_zephyr/connectedhomeip/examples/lighting-app/zephyr_boards$ ls src/
AppTask.cpp  ZclCallbacks.cpp  include  main.cpp
```

- includeの下にヘッダファイルを作成

```
~/matter_zephyr/connectedhomeip/examples/lighting-app/zephyr_boards$ ls src/include
AppTask.h  CHIPProjectConfig.h
```

ポーティング ～ ビルド成功までの道のり ～

- silabs, relateck, nrfconnectを参考に各ファイルの内容を書いていく
主に各ベンダーに依存しているところを避ける
例 **CMakeList.txt**のlistの記述

Silabs

```
list(APPEND ZEPHYR_EXTRA_MODULES ${CHIP_ROOT}/config/silabs)  
find_package(Zephyr HINTS $ENV{ZEPHYR_BASE})
```

```
# -Wmaybe-uninitialized has too many false positives, including on std::optional  
# and chip::Optional. Make it nonfatal.  
#
```

zephyr boards (今回のポーティング)

silabs -> zephyrへ変更

```
list(APPEND ZEPHYR_EXTRA_MODULES ${CHIP_ROOT}/config/zephyr/chip-module)  
list(APPEND ZEPHYR_EXTRA_MODULES ${CHIP_ROOT}/config/zephyr/)
```

```
find_package(Zephyr REQUIRED HINTS $ENV{ZEPHYR_BASE})
```

ここはもともとからある

```
~/matter_zephyr/connectedhomeip/config/zephyr$ ls  
Kconfig README.md app chip-gn chip-module ota-image.cmake ota-image_sysbuild.cmake zephyr-util.cmake  
~/matter_zephyr/connectedhomeip/config/zephyr$
```

ポーティング ～ ビルド成功までの道のり ～

- silabs, relateck, nrfconnectを参考に各ファイルの内容を書いていく
主に各ベンダーに依存しているところを避ける
例 **CMakeList.txt**のlistの記述

Silabs

```
list(APPEND ZEPHYR_EXTRA_MODULES ${CHIP_ROOT}/config/silabs)  
find_package(Zephyr HINTS $ENV{ZEPHYR_BASE})
```

```
# -Wmaybe-uninitialized has too many false positives, including on std::optional  
# and chip::Optional. Make it nonfatal.  
#
```

zephyr boards (今回のポーティング)

silabs -> zephyrへ変更

```
list(APPEND ZEPHYR_EXTRA_MODULES ${CHIP_ROOT}/config/zephyr/chip-module)  
list(APPEND ZEPHYR_EXTRA_MODULES ${CHIP_ROOT}/config/zephyr/)
```

```
find_package(Zephyr REQUIRED HINTS $ENV{ZEPHYR_BASE})
```

ここはもともとからある

```
~/matter_zephyr/connectedhomeip/config/zephyr$ ls  
Kconfig README.md app chip-gn chip-module ota-image.cmake ota-image_sysbuild.cmake zephyr-util.cmake  
~/matter_zephyr/connectedhomeip/config/zephyr$
```

ビルド

- `source zephyrproject/zephyr/zephyr-env.sh`

```
~$ source zephyrproject_matter/zephyr/zephyr-env.sh|
```

- `cd connectedhomeip` (`connectedhomeip`を展開したディレクトリ)
`source scripts/activate`

```
~$ cd matter_zephyr/connectedhomeip/  
~/matter_zephyr/connectedhomeip$ source scripts/activate.sh  
2026-03-27 13:01:28,176 Loading extra packages for 'linux'  
2026-03-27 13:01:28,176 Skipping: 'darwin' (i.e. /home/ihirata/ma
```

- `cd examples/lighting-app/zephyr_boards`
`west build -b nucleo_h563zi . -p`

```
~/matter_zephyr/connectedhomeip$ cd examples/lighting-app/zephyr_boards/  
~/matter_zephyr/connectedhomeip/examples/lighting-app/zephyr_boards$ west build -b nucleo_h563zi .
```


ビルド

Linkでundefined symbolが沢山でてくるので地道にgrepかけてシンボルの関数、メソッドを特定し、どうやったらコンパイル対象になるか調べる

特にmbedTLSのものが多かった。

対策としてprj.confにいろいろ定義していく。

```
CONFIG_MBEDTLS=y
CONFIG_MBEDTLS_BUILTIN=y
CONFIG_MBEDTLS_ECP_C=y
CONFIG_MBEDTLS_ECP_DP_SECP256R1_ENABLED=y
CONFIG_MBEDTLS_ECDH_C=y
CONFIG_MBEDTLS_ECDSA_C=y
#CONFIG_MBEDTLS_SHA256_C=y
#CONFIG_MBEDTLS_AES_C=y
#CONFIG_MBEDTLS_CCM_C=y
#CONFIG_MBEDTLS_CHACHA20_C=n
#CONFIG_MBEDTLS_POLY1305_C=n
```

```
CONFIG_MBEDTLS_PKCS5_C=y
CONFIG_MBEDTLS_HMAC_DRBG_C=y
#CONFIG_MBEDTLS_SHA256_C=y
CONFIG_MBEDTLS_MD_C=y
#CONFIG_MBEDTLS_CIPHER_C=y
CONFIG_POLL=y
CONFIG_EVENTS=y
CONFIG_REBOOT=y
CONFIG_MBEDTLS_CIPHER_AES_ENABLED=y
CONFIG_MBEDTLS_CTR_DRBG_C=y
CONFIG_MBEDTLS_HKDF_C=y
CONFIG_MBEDTLS_CIPHER_CCM_ENABLED=y
CONFIG_MBEDTLS_ENTROPY_C=y
```

FLASHメモリへ書き込み実行

ビルドが完了すると、いよいよボードへプログラムが書き込めます。
west flash コマンドを実行します。

```
connectedhomeip/examples/lighting-app/zephyr_boards$ west flash|
```

書き込みが完了すると書き込んだプログラムが実行されます....が
最初から綺麗には動きません。いろいろと調整が必要です。

動かない原因 リソース不足 スタックサイズ

スタックをまあまあ使います。

よく Stack overflowが発生して終了します。(スクリーンショットはなし)

prj.confに各スレッドのスタックサイズを設定するCONFIGを書きます。

```
CONFIG_NET_MGMT_EVENT_STACK_SIZE=4096  
#CONFIG_NET_MGMT_EVENT_STACK_SIZE=8192  
CONFIG_NET_RX_STACK_SIZE=4096  
#CONFIG_NET_RX_STACK_SIZE=8192  
CONFIG_NET_TX_STACK_SIZE=4096  
#CONFIG_NET_TX_STACK_SIZE=8192  
# CONFIG_NET_TCP_WORK_STACK_SIZE=2048 ない  
CONFIG_SYSTEM_WORKQUEUE_STACK_SIZE=4096  
#CONFIG_SYSTEM_WORKQUEUE_STACK_SIZE=8192  
CONFIG_NET_IP_STACK_PROTECTION
```

動かない原因 リソース不足 ヒープサイズ

スタックと同じで多めに取りました。

connetedhomeipでC++つかってる影響もあるかも

prj.confに設定します。

```
CONFIG_HEAP_MEM_POOL_SIZE=65536
```

動かない原因 リソース不足 その他

EVENTFDの数も足りませんでした。

```
# eventfd の最大数 (1 -> 4)  
CONFIG_ZVFS_EVENTFD_MAX=4  
  
# Poll/Select で同時に監視できる数 (3 -> 16)  
CONFIG_ZVFS_POLL_MAX=16
```

動かない原因 NVS

Non-Volatile Storage(NVS)も使います。(Matterの設定を保存します。)

```
CONFIG_SETTINGS=y  
CONFIG_SETTINGS_NVS=y  
CONFIG_NVS=y
```

Zephyrのソース

zephyr/boards/st/nucleo_h563zi (ここは変更しない)

boards/nucleo_h563zi.overlay

```
/ {  
  chosen {  
    /* NVSの保存先を明示的に指定 */  
    zephyr,settings-partition = &storage_partition;  
  };  
};
```

```
&flash0 {  
  partitions {  
    compatible = "fixed-partitions";  
    #address-cells = <1>;  
    #size-cells = <1>;  
  
    boot_partition: partition@0 {  
      label = "mcuboot";  
      reg = <0x00000000 DT_SIZE_K(64)>;  
    };  
  
    slot0_partition: partition@10000 {  
      label = "image-0";  
      reg = <0x00100000 DT_SIZE_K(960)>;  
    };  
  
    slot1_partition: partition@100000 {  
      label = "image-1";  
      reg = <0x00100000 DT_SIZE_K(960)>;  
    };  
  
    storage_partition: partition@1f0000 {  
      label = "storage";  
      reg = <0x001f0000 DT_SIZE_K(64)>;  
    };  
  };  
};
```

動かない原因 MDNS (MULTICAST DNS)

connectedhomeip/config/zephyr/chip-module/CMakeLists.txt

```
if(CONFIG_CHIP_WIFI)
    matter_add_gn_arg_string("chip_mdns" "minimal")
elseif(CONFIG_CHIP_ETHERNET)
    matter_add_gn_arg_string("chip_mdns" "minimal")
    matter_add_gn_arg_bool("chip_mdns_minimal" ON)
elseif(CONFIG_NET_L2_OPENTHREAD)
    matter_add_gn_arg_string("chip_mdns" "platform")
else()
    matter_add_gn_arg_string("chip_mdns" "none")
endif()
```

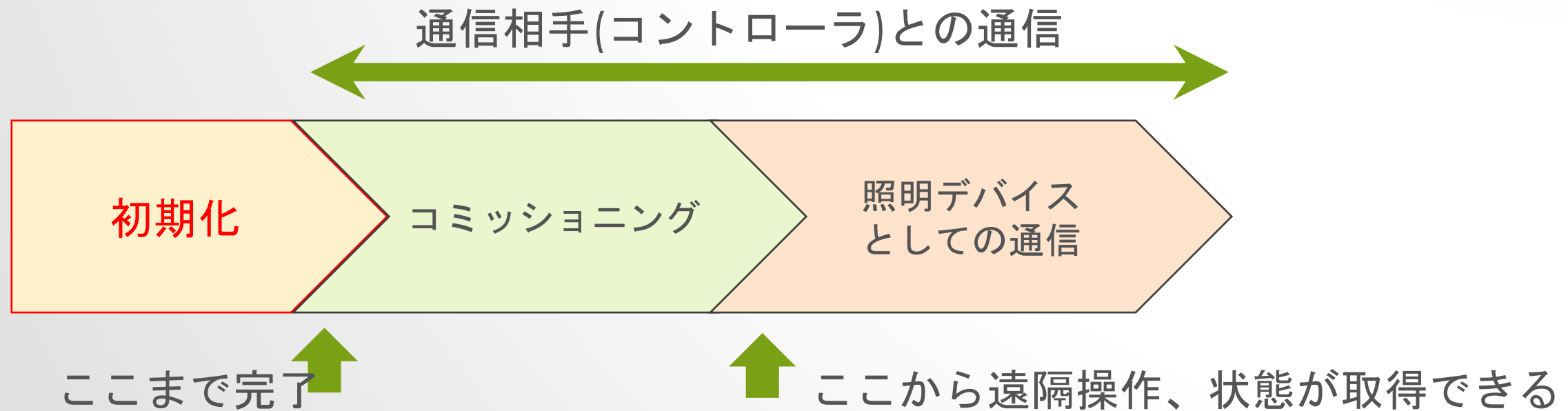
ここを有効にしたい
CONFIG__CHIP__ETHERNETがいる

ここが有効になっていた

prj.confへCONFIG_CHIP_ETHERNETを定義

```
CONFIG_CHIP_ETHERNET=y
CONFIG_CHIP_WIFI=n
```

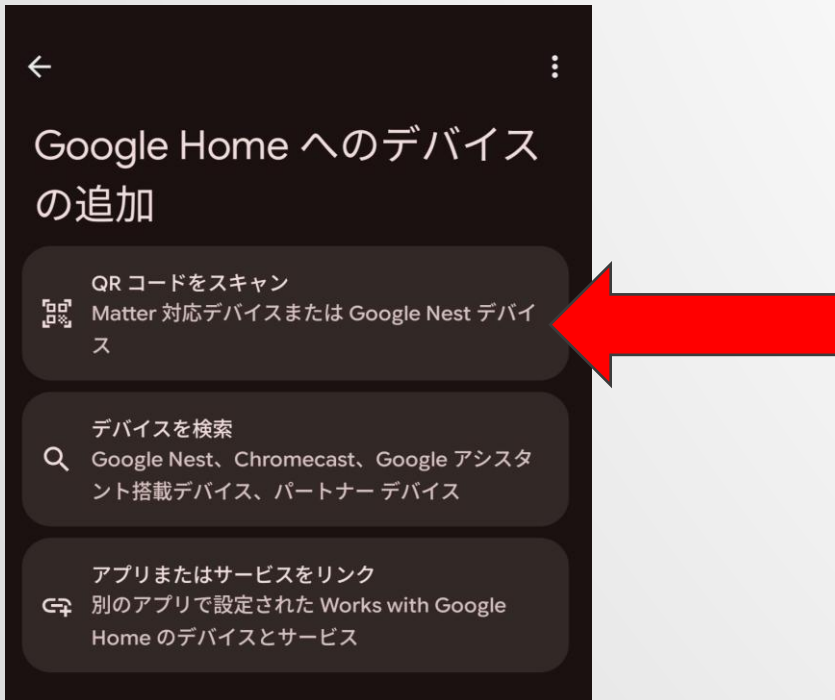
だいたい以上のようなかんじの
対策でいっけん動いているように
みえる、次は通信相手と通信し
てみます。



コミッショニング

開発したプログラムが動作しているボードをネットワーク(ファブリック-セキュア通信できる環境)へ登録します。

Google Homeからデバイス(ボード)を追加してみます。

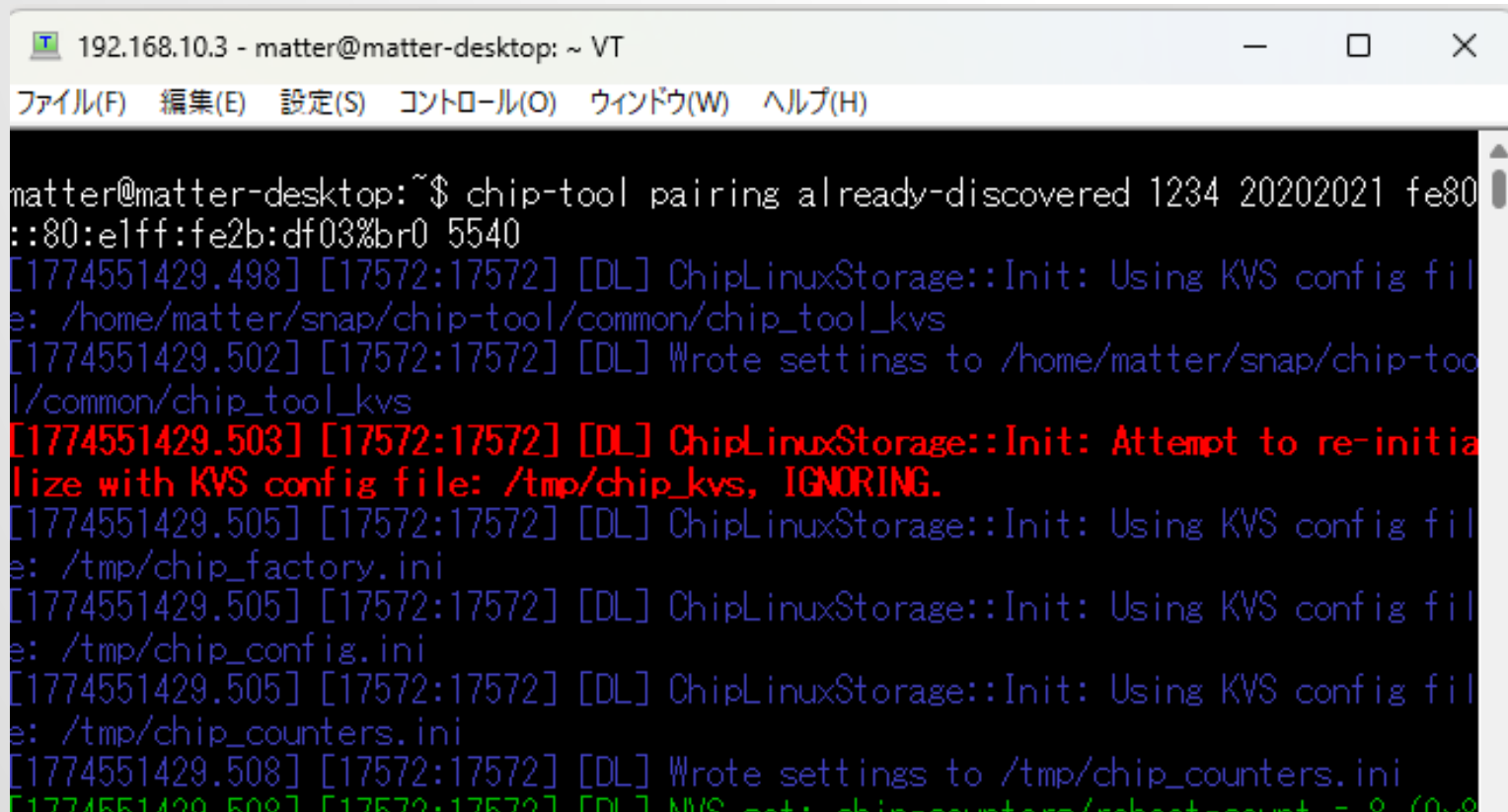


動きませんでした。

コミッショニング 通信相手を変更

Google Home、何故失敗したか出力してくれないので、
通信相手を変更

chip-toolというMatterをデバッグするためのツールを使いデバッグを継続



```
192.168.10.3 - matter@matter-desktop: ~ VT
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)
matter@matter-desktop:~$ chip-tool pairing already-discovered 1234 20202021 fe80
::80:e1ff:fe2b:df03%br0 5540
[1774551429.498] [17572:17572] [DL] ChipLinuxStorage::Init: Using KVS config file: /home/matter/snap/chip-tool/common/chip_tool_kvs
[1774551429.502] [17572:17572] [DL] Wrote settings to /home/matter/snap/chip-tool/common/chip_tool_kvs
[1774551429.503] [17572:17572] [DL] ChipLinuxStorage::Init: Attempt to re-initialize with KVS config file: /tmp/chip_kvs, IGNORING.
[1774551429.505] [17572:17572] [DL] ChipLinuxStorage::Init: Using KVS config file: /tmp/chip_factory.ini
[1774551429.505] [17572:17572] [DL] ChipLinuxStorage::Init: Using KVS config file: /tmp/chip_config.ini
[1774551429.505] [17572:17572] [DL] ChipLinuxStorage::Init: Using KVS config file: /tmp/chip_counters.ini
[1774551429.508] [17572:17572] [DL] Wrote settings to /tmp/chip_counters.ini
[1774551429.508] [17572:17572] [DL] NVS set: chip-counters/reboot-count = 8 (0x8)
```

コミッショニング 通信相手を変更

chip-tool
(コマンドラインツール)

Matter通信
(コミッショニング)



```
192.168.10.3 - matter@matter-desktop: ~ VT
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)
matter@matter-desktop:~$ chip-tool pairing already-discovered 1234 20202021 fe80
::80:e1ff:fe2b:df03%br0 5540
[1774551429.498] [17572:17572] [DL] ChipLinuxStorage::Init: Using KVS config fil
e: /home/matter/snap/chip-tool/common/chip_tool_kvs
[1774551429.502] [17572:17572] [DL] Wrote settings to /home/matter/snap/chip-to
ol/common/chip_tool_kvs
[1774551429.503] [17572:17572] [DL] ChipLinuxStorage::Init: Attempt to re-initia
lize with KVS config file: /tmp/chip_kvs, IGNORING.
[1774551429.505] [17572:17572] [DL] ChipLinuxStorage::Init: Using KVS config fil
e: /tmp/chip_factory.ini
[1774551429.505] [17572:17572] [DL] ChipLinuxStorage::Init: Using KVS config fil
e: /tmp/chip_config.ini
[1774551429.505] [17572:17572] [DL] ChipLinuxStorage::Init: Using KVS config fil
e: /tmp/chip_counters.ini
[1774551429.508] [17572:17572] [DL] Wrote settings to /tmp/chip_counters.ini
[1774551429.508] [17572:17572] [DL] NVS set: chip-counters/reboot-count = 8 (0x8
)
[1774551429.509] [17572:17572] [DL] Got Ethernet interface: eth0
[1774551429.509] [17572:17572] [DL] Found the primary Ethernet interface:eth0
```

```
COM4 - Tera Term VT
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)
[00:02:03.499,000] <dbg> chip: LogV: [EM]Flushed pending ack for MessageCounter:
121999655 on exchange 30482r
uart: $ [00:02:03.521,000] <inf> chip: [SC]Received Sigma3 msg
[00:02:03.521,000] <inf> chip: [SC]Received Sigma3 msg
uart: $ GetClock_RealTimeMS() src/platform/Zephyr/SystemTimeSupport.cpp 78
GetClock_RealTimeMS() src/platform/Zephyr/SystemTimeSupport.cpp 78
uart: $ GetClock_RealTime() src/platform/Zephyr/SystemTimeSupport.cpp 67
GetClock_RealTime() src/platform/Zephyr/SystemTimeSupport.cpp 67
uart: $ [00:02:03.565,000] <err> chip: [SC]The device does not support GetClock_
RealTimeMSO API: 3. Falling back to Last Known Good UTC Time
[00:02:03.565,000] <err> chip: [SC]The device does not support GetClock_RealTime
MSO API: 3. Falling back to Last Known Good UTC Time
uart: $ [00:02:05.467,000] <err> chip: [CR]mbedTLS error: -0xFFFF0010
[00:02:05.467,000] <err> chip: [CR]mbedTLS error: -0xFFFF0010
uart: $ [00:02:05.482,000] <err> chip: [SC]Failed to find valid cert during chain
traversal: 14
[00:02:05.482,000] <err> chip: [SC]Failed to find valid cert during chain traver
sal: 14
uart: $ [00:02:05.502,000] <err> chip: [DL]Long dispatch time: 1909 ms, for even
t type 2
[00:02:05.502,000] <err> chip: [DL]Long dispatch time: 1909 ms, for event type 2
uart: $ [00:02:05.520,000] <dbg> chip: LogV: [SC]Sending status report. Protocol
code 2, exchange 30482
[00:02:05.520,000] <dbg> chip: LogV: [SC]Sending status report. Protocol code 2,
```

1つ原因が分かった – CSRの生成

CSR (Certificate Signing Request)を生成できていない。
原因探ってみると、mbedtlsのコンフィグレーションで無効になっていた。

prj.confへMBEDTLSの定義を調整

```
CONFIG_MBEDTLS_X509_CSR_WRITE_C=y  
#CONFIG_MBEDTLS_PK_WRITE_C=y  
#CONFIG_MBEDTLS_PEM_WRITE_C=y  
CONFIG_MBEDTLS_ECDSA_DETERMINISTIC=y  
#CONFIG_MBEDTLS_X509_CRT_PARSE_C=y  
CONFIG_MBEDTLS_PK_WRITE_C=y
```

1つ原因が分かった – CSRの生成

ビルドかけてみるが....

丁寧にコードの中で依存関係を指摘してくれる。(コンパイルでエラーになる)

```
from /home/jhirata/zephyrproject_matter/modules/crypto/mbedtls/library/aes.c:14:
/home/jhirata/zephyrproject_matter/modules/crypto/mbedtls/include/mbedtls/check_config.h:463:2: error: #error "MBEDTLS_P
K_WRITE_C defined, but not all prerequisites"
 463 | #error "MBEDTLS_PK_WRITE_C defined, but not all prerequisites"
     | ^~~~~
/home/jhirata/zephyrproject_matter/modules/crypto/mbedtls/include/mbedtls/check_config.h:1031:2: error: #error "MBEDTLS_
X509_CREATE_C defined, but not all prerequisites"
1031 | #error "MBEDTLS_X509_CREATE_C defined, but not all prerequisites"
     | ^~~~~
```

```
#if defined(MBEDTLS_PK_WRITE_C) && \
    (!defined(MBEDTLS_ASN1_WRITE_C) || \
     !defined(MBEDTLS_OID_C) || \
     !defined(MBEDTLS_PK_C))
#error "MBEDTLS_PK_WRITE_C defined, but not all prerequisites"
#endif
```

1つ原因が分かった – CSRの生成

CONFIGでは定義できない、コンパイルオプション(-Dオプション)でダイレクトに定義する。

```
#if defined(MBEDTLS_PK_WRITE_C) && \
    (!defined(MBEDTLS_ASN1_WRITE_C) || \
     !defined(MBEDTLS_OID_C) || \
     !defined(MBEDTLS_PK_C))
#error "MBEDTLS_PK_WRITE_C defined, but not all prerequisites"
#endif
```

対策

prj.conf

```
#CONFIG_MBEDTLS_X509_CRT_PARSE_C=y
CONFIG_MBEDTLS_PK_WRITE_C=y
CONFIG_COMPILER_OPT="-DMBEDTLS_ASN1_WRITE_C -DMBEDTLS_OID_C -DMBEDTLS_PK_PARSE_C -DMBEDTLS_X509_CRT_PARSE_C"
```

これでこの件は解決！

2つ目の原因が分かった - 時刻

証明書の検証で時刻を使うはず。

正確な時刻をZephyr & その上のconnectedhomeipへ知らせていないのでは？。

```
COM4 - Tera Term VT
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)
[00:02:03.499,000] <dbg> chip: LogV: [EM]Flushed pending ack for MessageCounte
121999655 on exchange 30482r
art:~$ [00:02:03.521,000] <inf> chip: [SC]Received Sigma3 msg
[00:02:03.521,000] <inf> chip: [SC]Received Sigma3 msg
art:~$ GetClock_RealTimeMS() src/platform/Zephyr/SystemTimeSupport.cpp 78
GetClock_RealTimeMS() src/platform/Zephyr/SystemTimeSupport.cpp 78.....
art:~$ GetClock_RealTime() src/platform/Zephyr/SystemTimeSupport.cpp 67
GetClock_RealTime() src/platform/Zephyr/SystemTimeSupport.cpp 67
art:~$ [00:02:03.565,000] <err> chip: [SC]The device does not support GetClock
RealTimeMS() API: 3. Falling back to Last Known Good UTC Time
[00:02:03.565,000] <err> chip: [SC]The device does not support GetClock_RealTi
MS() API: 3. Falling back to Last Known Good UTC Time
art:~$ [00:02:05.467,000] <err> chip: [CR]mbedTLS error: -0xFFFF0010
[00:02:05.467,000] <err> chip: [CR]mbedTLS error: -0xFFFF0010
art:~$ [00:02:05.482,000] <err> chip: [SC]Failed to find valid cert during ch
n traversal: 14
[00:02:05.482,000] <err> chip: [SC]Failed to find valid cert during chain trav
sal: 14
art:~$ [00:02:05.502,000] <err> chip: [DL]Long dispatch time: 1909 ms, for ev
t type 2
```

こんなエラーが出てた

対策を考えてましたが、
本日を迎えました。

2つ目の原因が分かった - 時刻

証明書の検証で時刻を使うはず。

正確な時刻をZephyr & その上のconnectedhomeipへ知らせていない。

```
COM4 - Tera Term VT
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)
[00:02:03.499,000] <dbg> chip: LogV: [EM]Flushed pending ack for MessageCounte
[21999655 on exchange 30482r
jart:~$ [00:02:03.521,000] <inf> chip: [SC]Received Sigma3 msg
[00:02:03.521,000] <inf> chip: [SC]Received Sigma3 msg
jart:~$ GetClock_RealTimeMS() src/platform/Zephyr/SystemTimeSupport.cpp 78
GetClock_RealTimeMS() src/platform/Zephyr/SystemTimeSupport.cpp 78.....
jart:~$ GetClock_RealTime() src/platform/Zephyr/SystemTimeSupport.cpp 67
GetClock_RealTime() src/platform/Zephyr/SystemTimeSupport.cpp 67
jart:~$ [00:02:03.565,000] <err> chip: [SC]The device does not support GetClock
RealTimeMS() API: 3. Falling back to Last Known Good UTC Time
[00:02:03.565,000] <err> chip: [SC]The device does not support GetClock_RealTi
MS() API: 3. Falling back to Last Known Good UTC Time
jart:~$ [00:02:05.467,000] <err> chip: [CR]mbedtls error: -0xFFFF0010
[00:02:05.467,000] <err> chip: [CR]mbedtls error: -0xFFFF0010
jart:~$ [00:02:05.482,000] <err> chip: [SC]Failed to find valid cert during ch
n traversal: 14
[00:02:05.482,000] <err> chip: [SC]Failed to find valid cert during chain trav
sal: 14
jart:~$ [00:02:05.502,000] <err> chip: [DL]Long dispatch time: 1909 ms, for ev
t type: 2
[00:02:05.500,000] <err> chip: [EM]The device does not support GetClock_RealTimeMS() API: 3. Falling back to Last Known Good UTC Time
```

こんなエラーが出てた

コミッショニングのどこまで進んでいる？

Sigma3というメッセージが表示されています。

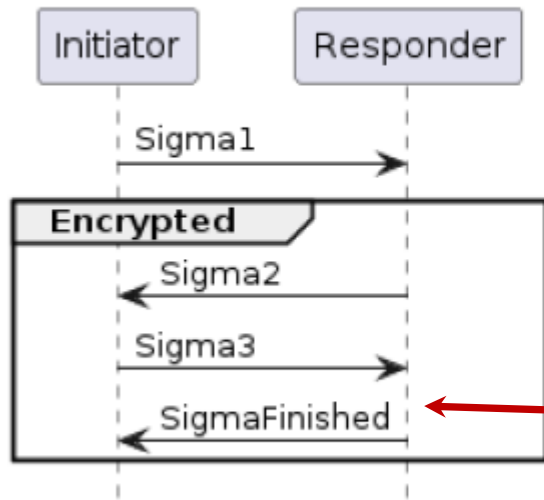
```
COM4 - Tera Term VT
ファイル(F) 編集(E) 設定(S) コントロール(O) ウィンドウ(W) ヘルプ(H)
[00:02:03.499,000] <dbg> chip: LogV: [EM]Flushed pending ack for MessageCounte
121999655 on exchange 30482r
jart:~$ [00:02:03.521,000] <inf> chip: [SC]received Sigma3 msg
[00:02:03.521,000] <inf> chip: [SC]Received Sigma3 msg
jart:~$ GetClock_RealTimeMSO src/platform/Zephyr/SystemTimeSupport.cpp 78
GetClock_RealTimeMSO src/platform/Zephyr/SystemTimeSupport.cpp 78
jart:~$ GetClock_RealTime() src/platform/Zephyr/SystemTimeSupport.cpp 67
GetClock_RealTime() src/platform/Zephyr/SystemTimeSupport.cpp 67
jart:~$ [00:02:03.565,000] <err> chip: [SC]The device does not support GetClock
RealTimeMSO API: 3. Falling back to Last Known Good UTC Time
[00:02:03.565,000] <err> chip: [SC]The device does not support GetClock_RealTi
MSO API: 3. Falling back to Last Known Good UTC Time
jart:~$ [00:02:05.467,000] <err> chip: [CR]MBEDTLS error: -0xFFFF0010
[00:02:05.467,000] <err> chip: [CR]MBEDTLS error: -0xFFFF0010
jart:~$ [00:02:05.482,000] <err> chip: [SC]Failed to find valid cert during ch
n traversal: 14
[00:02:05.482,000] <err> chip: [SC]Failed to find valid cert during chain trav
sal: 14
jart:~$ [00:02:05.502,000] <err> chip: [DL]Long dispatch time: 1909 ms, for ev
t type: 2
[00:02:05.500,000] <err> chip: [CR]MBEDTLS error: -0xFFFF0010
```



コミッショニングのどこまで進んでいる？

Matterの仕様書を確認

The basic protocol can be achieved within 2 round trips as shown below:

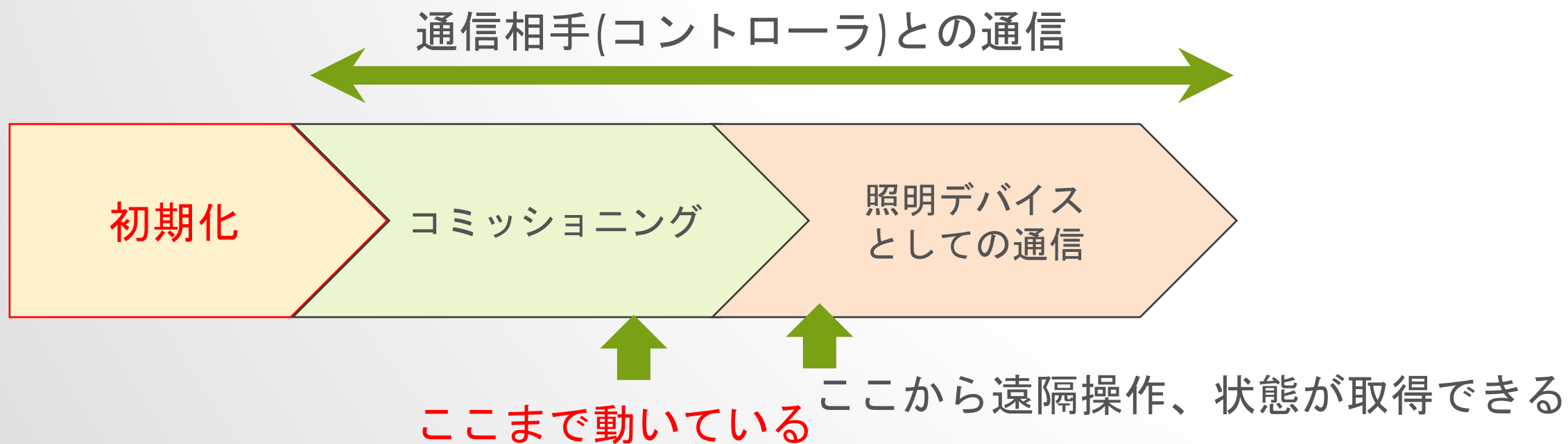


今このあたり

Figure 15. Basic Session Establishment

引用 : Matter Specification 1.3

全体で見ると



まとめ

- 結果的にNUCLEO H753ZIでMatterを動かせなかったが...
- 超えるべきハードルはいくつかあり、一つずつ解消していけばゴールへ近づいている実感はある。

目指しているゴール

家電の制御 (例 スマートスピーカーからの照明制御)



ご清聴ありがとうございました！