

Application Note

Enabling Matter on Sitara MPU



Krunal Bhargav and Randolph Sapp

ABSTRACT

This application note explores the implementation and usage of the Matter connectivity protocol on Sitara processor devices. The following sections outline the enablement and demonstration of Matter, including example data collected from SK-AM62B.

Table of Contents

1 Introduction	2
2 Current Implementation	2
3 Enablement	2
4 Demonstration	6
5 Summary	9
6 References	9

List of Figures

Figure 4-1. Hardware Setup.....	6
Figure 4-2. Creating an Endpoint.....	6
Figure 4-3. Expected Endpoint Log.....	7
Figure 4-4. Pairing With Endpoint Device.....	7
Figure 4-5. Successful Pairing.....	8
Figure 4-6. Setting Lock Status to Locked.....	8
Figure 4-7. Lock Status in Endpoint Log.....	9

Trademarks

All trademarks are the property of their respective owners.

1 Introduction

Matter is an open-source application-layer connectivity protocol that specializes in creating a uniform method of interacting with IoT devices. It's built on top of IP allowing it to work natively over multiple network standards, such as WiFi (802.11), Ethernet (802.3), and Thread (802.15.4).

2 Current Implementation

The most common implementation of this protocol is the reference implementation present in the chip-tool in the connectedhomeip project at: <https://github.com/project-chip/connectedhomeip>. This repository contains:

- An implementation of the Matter server
- A definition of the messaging interface
- All the required networking utils for broadcasting and listening for broadcast events, including:
 - A mDNS server
 - A DNS resolver
- Tools for enabling bluetooth provisioning
- A definition of every possible endpoint cluster type
- An example for every endpoint cluster
- An example of a Controller / Administrator application

There are only two things that are important for a simple demo: an Administrator and an Endpoint. As such, the focus will be on the chip-tool and lock-app examples. Starting with chip-tool, this example application has a Command Line Interface (CLI) that acts as an Administrator capable of linking to endpoints and issuing commands or fetching status based on the clusters enabled by that endpoint. The lock-app is an example of an endpoint that would normally be controlling an electronic latch. This application registers a handful of commands like:

- Lock
- Unlock
- Unbolt
- GetUser
- SetUser
- GetDoorState
- SetDoorState
- SetCredential
- GetCredential

Where each of these commands are registered with chiptool and have accompanying log and state change messages that are broadcast when called.

3 Enablement

For our demo, the SK-AM62B and for more information about the device are used, see the following link: <https://www.ti.com/tool/SK-AM62B>. With regards to software, the following steps may be used to compile the demo using Yocto:

1. For your Ubuntu host machine, download the prerequisites: https://software-dl.ti.com/processor-sdk-linux/esd/AM62X/09_00_00_03/exports/docs/linux/Overview_Building_the_SDK.html#prerequisites-one-time-setup.
2. `git clone https://git.ti.com/git/argo-project/oe-layersetup.git tisdsk`
3. `cd tisdsk`
4. `./oe-layertool-setup.sh -f configs/processor-sdk/processor-sdk-09.00.00-config.txt`
5. `cd sources`
6. `git clone -b kirkstone https://github.com/kraj/meta-clang.git`
7. `cd meta-argo/meta-argo-demos/recipes-apps`
8. `mkdir matter && cd matter`

9. Create a file called `matter_git.bb` and add the following content:

```
SUMMARY = "Matter IoT connectivity on TI boards"
DESCRIPTION = "This recipe primes the matter environment"
LICENSE = "Apache-2.0"
LIC_FILES_CHKSUM = "file://${COMMON_LICENSE_DIR}/
Apache-2.0;md5=89aea4e17d99a7cacdbeed46a0096b10"

BRANCH = "master"
SRC_URI = "git:github.com/project-chip/connectedhomeip.git;protocol=https;branch=${
{BRANCH}};lfs=1"

SRCREV = "a98bc64856aa161197e7dc7c1ffbdcc43323eda3"

do_matter_bootstrap[network] = "1"
do_compile[network] = "1"

TARGET_CC_ARCH += "${LDFLAGS}"
DEPENDS += " glib-2.0 gn-native ninja-native avahi dbus-glib-native pkgconfig-native python3-
native boost zap-native openssl-native ca-certificates-native clang-native"
RDEPENDS_${PN} += " libavahi-client openssl "
FILES:${PN} += "usr/share"
INSANE_SKIP:${PN} += "dev-so debug-deps strip"

PACKAGECONFIG ?= ""
PACKAGECONFIG[debug] = "is_debug=true,is_debug=false"

GN_TARGET_ARCH_NAME:aarch64 = "arm64"
GN_TARGET_ARCH_NAME:arm = "arm"
GN_TARGET_ARCH_NAME:x86 = "x86"
GN_TARGET_ARCH_NAME:x86-64 = "x64"

def gn_target_arch_name(d):
    """Returns a GN architecture name corresponding to the target machine's
    architecture."""
    name = d.getVar("GN_TARGET_ARCH_NAME")
    if name is None:
        bb.fatal('Unsupported target architecture. A valid override for the '
        'GN_TARGET_ARCH_NAME variable could not be found.')
    return name

# this variable must use spaces and double quotes for parameter strings because
# *gn* is evil
GN_ARGS = " \
    ${PACKAGECONFIG_CONFARGS} \
    target_cpu="${@gn_target_arch_name(d)}" \
    target_arch="${TUNE_FEATURES}" \
    target_os="linux" \
    treat_warnings_as_errors=false \
    enable_rtti=true \
    enable_exceptions=true \
"

# Make sure pkg-config, when used with the host's toolchain to build the
# binaries we need to run on the host, uses the right pkg-config to avoid
# passing include directories belonging to the target.
GN_ARGS += 'host_pkg_config="pkg-config-native"'

s = "${WORKDIR}/git"

common_configure() {
    # this block must use spaces and double quotes for strings because *gn* is
    # evil
    PKG_CONFIG_SYSROOT_DIR=${PKG_CONFIG_SYSROOT_DIR} \
    PKG_CONFIG_LIBDIR=${PKG_CONFIG_PATH} \
    gn gen out/ --args='
        ${GN_ARGS}
        import("//build_overrides/build.gni")
        target_cflags=[
            "-DCHIP_DEVICE_CONFIG_WIFI_STATION_IF_NAME=\"wlan0\"",
            "-DCHIP_DEVICE_CONFIG_LINUX_DHCP_CMD=\"udhcp -b -i %s \"",
        ]
        custom_toolchain="${build_root}/toolchain/custom"
        target_cc="${CC}"
        target_cxx="${CXX}"
        target_ar="${AR}"
    '
}

```

```

export https_proxy
export http_proxy
export ftp_proxy
export no_proxy

do_matter_bootstrap() {
    . ${S}/scripts/bootstrap.sh
}

do_configure() {
    . scripts/activate.sh
    pip install click

    cd ${S}/examples/chip-tool
    common_configure

    cd ${S}/examples/lock-app/linux
    common_configure

    cd ${S}/examples/thermostat/linux
    common_configure

    cd ${S}/examples/lighting-app/linux
    common_configure
}

do_compile() {
    . scripts/activate.sh

    cd ${S}/examples/chip-tool
    ninja -C out/

    cd ${S}/examples/lock-app/linux
    ninja -C out/

    cd ${S}/examples/thermostat/linux
    ninja -C out/

    cd ${S}/examples/lighting-app/linux
    ninja -C out/
}

do_install() {
    install -d -m 755 ${D}${bindir}

    # Install chip-tool
    install ${S}/examples/chip-tool/out/chip-tool ${D}${bindir}

    # lock-app
    install ${S}/examples/lock-app/linux/out/chip-lock-app ${D}${bindir}
    install ${S}/examples/thermostat/linux/out/thermostat-app ${D}${bindir}
    install ${S}/examples/lighting-app/linux/out/chip-lighting-app ${D}${bindir}
}

addtask matter_bootstrap after do_unpack before do_configure

INSANE_SKIP_${PN} = "ldflags"

```

10. Create a file called `zap_git.bb` and add the following content:

```

PN = "zap-native"
SUMMARY = "ZAP prebuilt tools"
DESCRIPTION = "ZAP prebuilt binaries"
LICENSE = "Apache-2.0"
LIC_FILES_CHKSUM = "file://${COMMON_LICENSE_DIR}/
Apache-2.0;md5=89aea4e17d99a7cacdbeed46a0096b10"

PACKAGES = "${PN}"

PV = "v2023.08.04-nightly"
SRC_URI = "https://github.com/project-chip/zap/releases/download/${PV}/zap-linux-
x64.zip;unpack=yes"
SRC_URI[sha256sum] = "b254a0c066ef6b1fe7c2bdd1ab5b137ca80413f0952dfe6e64f4b0fdc4479b55"

S = "${WORKDIR}"

#INSANE_SKIP:${PN} = " already-stripped arch file-rdeps "

```

```

BBCLASSEXTEND = "native"
INHIBIT_PACKAGE_STRIP = "1"
INHIBIT_SYSROOT_STRIP = "1"
INHIBIT_PACKAGE_DEBUG_SPLIT = "1"
INHIBIT_FILE_RDEPS = "1"
INHIBIT_PACKAGE_DEBUG_SPLIT_CHECK = "1"
INHIBIT_PACKAGE_DEPMODE_CHECK = "1"
INHIBIT_PACKAGE_RELOCATE = "1"
INHIBIT_PACKAGE_UNPACK = "1"

INSANE_SKIP:${PN} += "dev-so"
inherit native

do_install() {
    install -d -m 0755 ${D}${bindir}/
    cp -ar zap* ${D}${bindir}/
    # This is a workaround to bypass the issue that zap-cli modified by build system
    chmod 444 ${D}${bindir}/zap-cli
}

do_package_qa[noexec] = "1"
EXCLUDE_FROM_SHLIBS = "1"

# This is a workaround to bypass the issue that zap-cli modified by build system
do_deploy() {
    chmod 755 ${D}${bindir}/zap-cli
}

do_populate_sdk:append() {
    chmod 755 ${D}${bindir}/zap-cli
}

addtask deploy after do_install do_populate_sysroot
addtask deploy before do_cleansstate
addtask deploy before do_clean

```

11. cd ../../../../

12. cd sources/bitbake/lib/bb/fetch2/

13. Modify gitism.py like the following:

```

diff --git a/lib/bb/fetch2/gitism.py b/lib/bb/fetch2/gitism.py
index c5f7c03c..ee852224 100644
--- a/lib/bb/fetch2/gitism.py
+++ b/lib/bb/fetch2/gitism.py
@@ -122,6 +122,7 @@ class GitSM(Git):
     url += ';protocol=%s' % proto
     url += ";name=%s" % module
     url += ";subpath=%s" % module
+    url += ";lfs=1"

    ld = d.createCopy()
    # Not necessary to set SRC_URI, since we're passing the URI to
@@ -238,7 +239,7 @@ class GitSM(Git):
    # All submodules should already be downloaded and configured in the tree. This simply sets
    # up the configuration and checks out the files. The main project config should remain
    # unmodified, and no download from the internet should occur.
-    runfetchcmd("%s submodule update --recursive --no-fetch" % (ud.basecmd), d, quiet=True,
workdir=ud.destdir)
+    runfetchcmd("GIT_LFS_SKIP_SMUDGE=1 %s submodule update --recursive --no-fetch" %
(ud.basecmd), d, quiet=True, workdir=ud.destdir)

def implicit_urldata(self, ud, d):
    import shutil, subprocess, tempfile

```

14. cd ../../../../build/

15. Open file conf/local.conf and add the following at the bottom of file: IMAGE_INSTALL:append = "matter"

16. . conf/setenv

17. MACHINE=am62xx-evm bitbake-layers add-layer ../sources/meta-clang/

18. MACHINE=am62xx-evm bitbake tisdk-default-image

19. Burn SD card using the wic image generated in the following directory: ./arago-tmp-default-glibc/deploy/images/am62xx-evm/tisdk-default-image-am62xx-evm.wic.xz

After generating the wic image, see the following instructions for booting the EVM with SD card: https://dev.ti.com/tirex/content/tirex-product-tree/am62x-devtools/docs/am62x_skevm_quick_start_guide.html

4 Demonstration

Figure 4-1 shows two AM62x devices using the chip-tool and lock-app interfacing with each other over Ethernet.

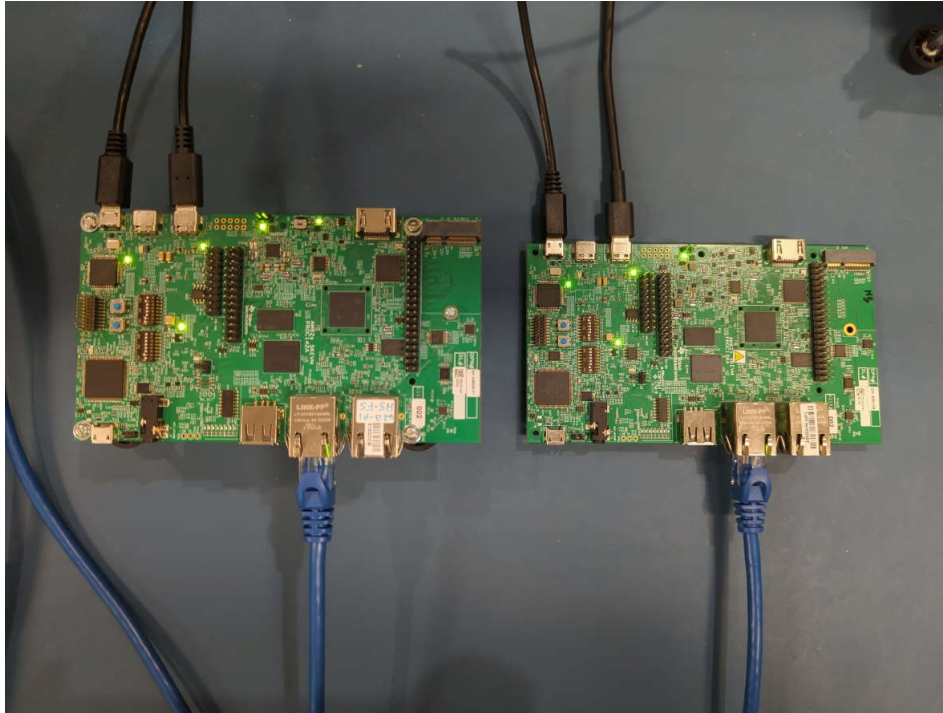


Figure 4-1. Hardware Setup

Figure 4-2 shows how to setup the AM62x device as an endpoint using the lock app.

```
[ #0 Endpoint ]  
root@am62xx-evm:~# /usr/bin/chip-lock-app  
|
```

Figure 4-2. Creating an Endpoint

Figure 4-3 shows what an expected endpoint log should look like. Note the device configuration information.

```
[ # Endpoint ]
[1698091073.104559][7891:7891] CHIP-DIS: Responding with DD45B3C0D6209788_matter_udp.local
[1698091073.104577][7891:7891] CHIP-DIS: CHIP minimal mDNS configured as 'Commissionable node device'; instance name
: DD45B3C0D6209788
[1698091073.111691][7891:7891] CHIP-DIS: Responding with 1053490F614A0800.local
[1698091073.111773][7891:7891] CHIP-DIS: Updating services using commissioning mode 1
[1698091073.115639][7891:7891] CHIP-DIS: CHIP minimal mDNS started advertising.
[1698091073.130895][7891:7891] CHIP-DIS: Using wifi MAC for hostname
[1698091073.130268][7891:7891] CHIP-DIS: Advertise commission parameter vendorID=65521 productID=32769 discriminator
=3840/16 cm=1
[1698091073.130320][7891:7891] CHIP-DIS: Responding with _matter_udp.local
[1698091073.130336][7891:7891] CHIP-DIS: Responding with DD45B3C0D6209788_matter_udp.local
[1698091073.130351][7891:7891] CHIP-DIS: Responding with 1053490F614A0800.local
[1698091073.130365][7891:7891] CHIP-DIS: Responding with 1053490F614A0800.local
[1698091073.130384][7891:7891] CHIP-DIS: Responding with _V65521_sub_matter_udp.local
[1698091073.130400][7891:7891] CHIP-DIS: Responding with _S15_sub_matter_udp.local
[1698091073.130419][7891:7891] CHIP-DIS: Responding with _3840_sub_matter_udp.local
[1698091073.130436][7891:7891] CHIP-DIS: Responding with _CM_sub_matter_udp.local
[1698091073.130461][7891:7891] CHIP-DIS: Responding with DD45B3C0D6209788_matter_udp.local
[1698091073.130476][7891:7891] CHIP-DIS: CHIP minimal mDNS configured as 'Commissionable node device'; instance name
: DD45B3C0D6209788
[1698091073.135851][7891:7891] CHIP-DIS: mDNS service published: _matter_udp
[1698091073.135989][7891:7891] CHIP-DIS: CASE Server enabling CASE session setups
[1698091073.136108][7891:7891] CHIP-IN: SecureSession[0xaaadbd9549]: Allocated Type:2 LSIID:68447
[1698091073.136141][7891:7891] CHIP-SEC: Allocated SecureSession (0xaaadbd9549) - waiting for Signal msg
[1698091073.136164][7891:7891] CHIP-SVR: Joining Multicast groups
[1698091073.136188][7891:7891] CHIP-ZCL: Emitting Startup event
[1698091073.136200][7891:7891] CHIP-EVL: LogEvent event number: 0x0000000000000002 priority: 2, endpoint id: 0x0 c
acter id: 0x00000002 event id: 0x0 Epoch timestamp: 0x0000010B5E1D7E70
[1698091073.136270][7891:7891] CHIP-SVR: Server initialization complete
[1698091073.136306][7891:7891] CHIP-SVR: Server Listening...
[1698091073.136320][7891:7891] CHIP-DL: Device Configuration:
[1698091073.136365][7891:7891] CHIP-DL: Serial Number: TEST_SN
[1698091073.136431][7891:7891] CHIP-DL: Vendor Id: 65521 (0xFF1)
[1698091073.136458][7891:7891] CHIP-DL: Product Id: 32769 (0x8001)
[1698091073.136474][7891:7891] CHIP-DL: Product Name: TEST_PRODUCT
[1698091073.136496][7891:7891] CHIP-DL: Hardware Version: 0
[1698091073.136510][7891:7891] CHIP-DL: Setup Pin Code (0 for UNKNOWN/ERROR): 20202021
[1698091073.136525][7891:7891] CHIP-DL: Setup Discriminator (0xFFFF for UNKNOWN/ERROR): 3840 (0xF00)
[1698091073.136540][7891:7891] CHIP-DL: Manufacturing Date: (not set)
[1698091073.136559][7891:7891] CHIP-DL: Device Type: 65535 (0xFFFF)
[1698091073.136583][7891:7891] CHIP-SVR: SetupQRCode: [MT:-24J042C08KA0648000]
[1698091073.136605][7891:7891] CHIP-SVR: Copy/paste the below URL in a browser to see the QR Code:
[1698091073.466833][7891:7891] CHIP-SVR: https://project-chip.github.io/connectedhomeip/qrcode.html?data=MT%3A-24J04
2C08KA0648000
[1698091073.467126][7891:7891] CHIP-SVR: Manual pairing code: [34970112332]
[1698091073.479352][7891:7891] CHIP-DL: TRACE: Bus acquired for name MATTER-3840
[1698091073.479820][7891:7891] CHIP-DL: CREATE service object at /chipoble/led3/service
[1698091073.481740][7891:7891] CHIP-DL: Create characteristic object at /chipoble/led3/service/c1
[1698091073.483175][7891:7891] CHIP-DL: Create characteristic object at /chipoble/led3/service/c2
[1698091073.483983][7891:7891] CHIP-DL: CHIP BTP C1 /chipoble/led3/service
[1698091073.484023][7891:7891] CHIP-DL: CHIP BTP C2 /chipoble/led3/service
[1698091073.484038][7891:7891] CHIP-DL: CHIP_ENABLE_ADDITIONAL_DATA_ADVERTISING is FALSE
[Admin] 0:~# - 1:~# Matter Demo*
```

Figure 4-3. Expected Endpoint Log

Figure 4-4 shows how an administrator would pair with the endpoint using the chip-tool.

```
[ #1 Administrator ]
root@am62xx-evm:~# /usr/bin/chip-tool pairing onnetwork 1 20202021
```

Figure 4-4. Pairing With Endpoint Device

Figure 4-5 shows the expect log of a successful pairing attempt. Note the CommissioningComplete response in the log.

```

[ # Administrator ]
[1698091101.606158][7889-7892] CHIP:DL: HandlePlatformSpecificBLEEvent 32793
[1698091101.613771][7889-7892] CHIP:EM: >>> [E:575771 S:16952 M:254685401 (Ack:55486435)] (S) Msg RX from 1:0000000000000000 [5361] --- Type 0001:09 (LM:InvokeCommandResponse)
[1698091101.614240][7889-7892] CHIP:DM: Found matching exchange: 575771, Delegate: 0xfffffc01ab58
[1698091101.615089][7889-7892] CHIP:EM: Rxd Ack, Removing MessageCounter:55486435 from Retrans Table on exchange 575771
[1698091101.613966][7889-7892] CHIP:DMG: ICR moving to [Response]
[1698091101.614096][7889-7892] CHIP:DMG: Received Command Response Data, Endpoint=0 Cluster=0x0000_0030 Command=0x00_00_0005
[1698091101.614155][7889-7892] CHIP:CTL: Received CommissioningComplete response, errorCode=0
[1698091101.614201][7889-7892] CHIP:CTL: Successfully finished commissioning step 'SendComplete'
[1698091101.614240][7889-7892] CHIP:CTL: Commissioning stage next step: 'SendComplete' -> 'Cleanup'
[1698091101.614286][7889-7892] CHIP:CTL: Performing next commissioning step 'Cleanup'
[1698091101.614329][7889-7892] CHIP:IN: SecureSession[0xfffffc00ddc0]: MarkForEviction Type:1 LSID:16951
[1698091101.614368][7889-7892] CHIP:SC: SecureSession[0xfffffc00ddc0, LSID:16951]: State change 'kActive' -> 'kPendingEviction'
[1698091101.614556][7889-7892] CHIP:IN: SecureSession[0xfffffc00ddc0]: Released - Type:1 LSID:16951
[1698091101.614612][7889-7892] CHIP:CTL: Successfully finished commissioning step 'Cleanup'
[1698091101.614637][7889-7892] CHIP:T00: Device commissioning completed with success
[1698091101.614933][7889-7892] CHIP:DMG: ICR moving to [Waiting]
[1698091101.614937][7889-7892] CHIP:EM: <<< [E:575771 S:16952 M:55486436 (Ack:254685401)] (S) Msg TX to 1:0000000000000000 [5361] --- Type 0000:10 (SecureChannel:StandaloneAck)
[1698091101.615084][7889-7892] CHIP:IN: (S) Sending msg 55486436 on secure session with LSID: 16952
[1698091101.615224][7889-7892] CHIP:EM: Flushed pending ack for MessageCounter:254685401 on exchange 575771
[1698091101.617188][7889-7892] CHIP:DL: HandlePlatformSpecificBLEEvent 32793
[1698091101.620558][7889-7889] CHIP:CTL: Shutting down the commissioner
[1698091101.620593][7889-7889] CHIP:CTL: Stopping commissioning discovery over DNS-D
[1698091101.620711][7889-7889] CHIP:CTL: Shutting down the controller
[1698091101.620727][7889-7889] CHIP:IN: Expiring all sessions for fabric 0x1!
[1698091101.620739][7889-7889] CHIP:IN: SecureSession[0xfffffc03e130]: MarkForEviction Type:2 LSID:16952
[1698091101.620753][7889-7889] CHIP:SC: SecureSession[0xfffffc03e130, LSID:16952]: State change 'kActive' -> 'kPendingEviction'
[1698091101.620769][7889-7889] CHIP:IN: SecureSession[0xfffffc03e130]: Released - Type:2 LSID:16952
[1698091101.620786][7889-7889] CHIP:FP: Forgetting fabric 0x1
[1698091101.620811][7889-7889] CHIP:TS: Pending Last Known Good Time: 2023-08-10T21:07:57
[1698091101.621809][7889-7889] CHIP:TS: Previous Last Known Good Time: 2023-08-10T21:07:57
[1698091101.621823][7889-7889] CHIP:TS: Reverted Last Known Good Time to previous value
[1698091101.621876][7889-7889] CHIP:CTL: Shutting down the commissioner
[1698091101.621882][7889-7889] CHIP:CTL: Stopping commissioning discovery over DNS-D
[1698091101.621937][7889-7889] CHIP:CTL: Shutting down the controller
[1698091101.949699][7889-7889] CHIP:CTL: Shutting down the System State, this will teardown the CHIP Stack
[1698091101.950263][7889-7889] CHIP:DMG: All ReadHandler-s are clean, clear GlobalDirtySet
[1698091101.950309][7889-7889] CHIP:FP: Shutting down FabricTable
[1698091101.950445][7889-7889] CHIP:TS: Pending Last Known Good Time: 2023-08-10T21:07:57
[1698091101.950648][7889-7889] CHIP:TS: Previous Last Known Good Time: 2023-08-10T21:07:57
[1698091101.950674][7889-7889] CHIP:TS: Reverted Last Known Good Time to previous value
[1698091101.951042][7889-7889] CHIP:DL: writing settings to file (/tmp/chip_counters.ini--0RV0bv)
[1698091101.951372][7889-7889] CHIP:DL: renamed tmp file to file (/tmp/chip_counters.ini)
[1698091101.951637][7889-7889] CHIP:DL: NWS set: chip-counters/total-operational-hours = 0 (0x0)
[1698091101.951694][7889-7889] CHIP:DL: Inet Layer shutdown
[1698091101.951718][7889-7889] CHIP:DL: BLE shutdown
[1698091101.951741][7889-7889] CHIP:DL: System Layer shutdown
root@am62xx-evm-#
Administrator 14:58 23-Oct-23

```

Figure 4-5. Successful Pairing

Figure 4-6 shows that the status of the endpoint is set to be locked.

```

[ # Administrator ]
[1698091101.951741][7889-7889] CHIP:DL: System Layer shutdown
root@am62xx-evm-# /usr/bin/chip-tool doorLock lock-door 1 1 --timedInteractionTimeoutMs 1000
[1698091110.132191][7954-7954] CHIP:DL: ChpLinuxStorage::Init: Using KVS config file: /tmp/chip_kv
[1698091110.137725][7954-7954] CHIP:DL: ChpLinuxStorage::Init: Using KVS config file: /tmp/chip_factory.ini
[1698091110.138620][7954-7954] CHIP:DL: ChpLinuxStorage::Init: Using KVS config file: /tmp/chip_config.ini
[1698091110.138892][7954-7954] CHIP:DL: ChpLinuxStorage::Init: Using KVS config file: /tmp/chip_counters.ini
[1698091110.138375][7954-7954] CHIP:DL: renamed tmp file to file (/tmp/chip_counters.ini-5015k)
[1698091110.138656][7954-7954] CHIP:DL: renamed tmp file to file (/tmp/chip_counters.ini)
[1698091110.138685][7954-7954] CHIP:DL: NWS set: chip-counters/reboot-count = 2 (0x2)
[1698091110.138681][7954-7954] CHIP:DL: Got Ethernet interface: eth0
[1698091110.139219][7954-7954] CHIP:DL: Found the primary Ethernet interface: eth0
[1698091110.140130][7954-7954] CHIP:DL: Failed to get WiFi interface
[1698091110.140151][7954-7954] CHIP:DL: Failed to reset WiFi statistic counts
[1698091110.140210][7954-7954] CHIP:IN: UDP: Init bind&listen port=0
[1698091110.140221][7954-7954] CHIP:IN: UDP: Init bound to port=54854
[1698091110.140335][7954-7954] CHIP:IN: UDP: Init bind&listen port=0
[1698091110.140389][7954-7954] CHIP:IN: UDP: Init bound to port=43943

```

Figure 4-6. Setting Lock Status to Locked

Figure 4-7 shows the status reported on the endpoint following the lock-door request.

```

[ #0 Endpoint ]
[1698091110.564418][7891.7891] CHIP-DL: HandlePlatformSpecificBLEEvent 32793
[1698091110.567324][7891.7891] CHIP-EM: >>> [E:36465r S:60448 M:92270786] (S) Msg RX from 1:00000000018669 [6361]
--- Type 0001:00 (IM:InvokeRequest)
[1698091110.567602][7891.7891] CHIP-EM: Handling via exchange: 36465r, Delegate: @xa0aaabdcf3f60
[1698091110.567667][7891.7891] CHIP-DMG: Got Timed Request with timeout 1000: handler @xa0aaabdcf3f60 exchange 36465r
[1698091110.567944][7891.7891] CHIP-EM: <<< [E:36465r S:60448 M:247907961 (Ack:92270786)] (S) Msg TX to 1:0000000000
018669 [6361] --- Type 0001:00 (IM:StatusResponse)
[1698091110.568073][7891.7891] CHIP-EM: (S) Sending msg 247907961 on secure session with LSID: 60448
[1698091110.568574][7891.7891] CHIP-DMG: Timed Request Time Limit 0x000000000033588F: handler @xa0aaabdcf3f60 exchang
e 36465r
[1698091110.568699][7891.7891] CHIP-DL: HandlePlatformSpecificBLEEvent 32793
[1698091110.568978][7891.7891] CHIP-EM: >>> [E:36464r S:0 M:27467073 (Ack:89265551)] (U) Msg RX from 0:A347E54502CEE
B4B [0080] --- Type 0080:10 (SecureChannel:StandaloneAck)
[1698091110.569096][7891.7891] CHIP-EM: Found matching exchange: 36464r, Delegate: (nil)
[1698091110.569308][7891.7891] CHIP-EM: Rxd Ack; Removing MessageCounter:89265551 from Retrans Table on exchange 364
64r
[1698091110.569421][7891.7891] CHIP-DL: HandlePlatformSpecificBLEEvent 32793
[1698091110.598427][7891.7891] CHIP-EM: >>> [E:36465r S:60448 M:92270787 (Ack:247907961)] (S) Msg RX from 1:00000000
00018669 [6361] --- Type 0001:00 (IM:InvokeCommandRequest)
[1698091110.598491][7891.7891] CHIP-EM: Found matching exchange: 36465r, Delegate: @xa0aaabdcf3f60
[1698091110.598497][7891.7891] CHIP-EM: Rxd Ack; Removing MessageCounter:247907961 from Retrans Table on exchange 36
465r
[1698091110.598517][7891.7891] CHIP-DMG: Timed following action arrived at 0x0000000000339921: handler @xa0aaabdcf3f6
0 exchange 36465r
[1698091110.598531][7891.7891] CHIP-DMG: Handling timed invoke to IM engine: handler @xa0aaabdcf3f60 exchange 36465r
[1698091110.598606][7891.7891] CHIP-DMG: Received command for Endpoint=1 Cluster=0x0000_0101 Command=0x0000_0000
[1698091110.598649][7891.7891] CHIP-ZCL: Received command: LockDoor
[1698091110.598720][7891.7891] CHIP-ZCL: Door Lock App: PID code is not specified [endpointId=1]
[1698091110.598757][7891.7891] CHIP-ZCL: Door Lock App: setting door lock state to "Locked" [endpointId=1]
[1698091110.598817][7891.7891] CHIP-DMG: Command handler moving to [Preparing]
[1698091110.598864][7891.7891] CHIP-DMG: Command handler moving to [AddingComm]
[1698091110.598905][7891.7891] CHIP-DMG: Command handler moving to [AddedComm]
[1698091110.598961][7891.7891] CHIP-DMG: Decreasing reference count for CommandHandler, remaining 0
[1698091110.599087][7891.7891] CHIP-EM: <<< [E:36465r S:60448 M:247907962 (Ack:92270787)] (S) Msg TX to 1:0000000000
018669 [6361] --- Type 0001:00 (IM:InvokeCommandResponse)
[1698091110.599102][7891.7891] CHIP-EM: (S) Sending msg 247907962 on secure session with LSID: 60448
[1698091110.599479][7891.7891] CHIP-DMG: Command handler moving to [CommandSent]
[1698091110.599522][7891.7891] CHIP-DMG: Command handler moving to [AwaitingDe]
[1698091110.599579][7891.7891] CHIP-DL: HandlePlatformSpecificBLEEvent 32793
[1698091110.599641][7891.7891] CHIP-DL: HandlePlatformSpecificBLEEvent 32793
[1698091110.599684][7891.7891] CHIP-DL: writing settings to file (/tmp/chip_kvz-p8gmVq)
[1698091110.598655][7891.7891] CHIP-DL: renamed tmp file to file (/tmp/chip_kvz)
[1698091110.900748][7891.7891] CHIP-DMG: Endpoint 1, Cluster 0x0000_0101 update version to 1c9fd231
[1698091110.900789][7891.7891] CHIP-ZCL: Door Lock attribute changed
[1698091110.900881][7891.7891] CHIP-EVL: LogEvent event number: 0x0000000000000004 priority: 2, endpoint id: 0x1 c1
uster id: 0x0000_0101 event id: 0x2 Epoch timestamp: 0x000001885E1E11F4
[1698091110.901306][7891.7891] CHIP-EM: >>> [E:36465r S:60448 M:92270788 (Ack:247907962)] (S) Msg RX from 1:00000000
00018669 [6361] --- Type 0080:10 (SecureChannel:StandaloneAck)
[1698091110.901375][7891.7891] CHIP-EM: Found matching exchange: 36465r, Delegate: (nil)
[1698091110.901431][7891.7891] CHIP-EM: Rxd Ack; Removing MessageCounter:247907962 from Retrans Table on exchange 36
465r
[1698091110.901491][7891.7891] CHIP-DL: HandlePlatformSpecificBLEEvent 32793
[Admin] 0:~bash-1.Matter Demo*
  
```

Figure 4-7. Lock Status in Endpoint Log

To see a recorded demonstration of the above with full endpoint and administrator logs being updated in sync, see the following: <https://asciinema.org/a/620956>.

5 Summary

The main goal of this application note is to demonstrate how to compile a reference implementation of matter from the connectedhomeip project and run a simple lock/unlock demo. Even though a AM62x device is used, the above instructions are applicable to any ARM32 bit and ARM64-bit TI processors.

6 References

- Texas Instruments, [AM625](#), product folder.

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES "AS IS" AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI's products are provided subject to [TI's Terms of Sale](#) or other applicable terms available either on [ti.com](https://www.ti.com) or provided in conjunction with such TI products. TI's provision of these resources does not expand or otherwise alter TI's applicable warranties or warranty disclaimers for TI products.

TI objects to and rejects any additional or different terms you may have proposed.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2024, Texas Instruments Incorporated