Getting started with MW320 in Matter
1 INTRODUCTION

Matter (previously known as Project CHIP) is a new single, unified, application-layer connectivity standard designed to enable developers to connect and build reliable, secure IoT ecosystems and increase compatibility among Smart Home and Building devices.

For enabling Matter devices, NXP offers scalable, flexible and secure platforms to enable the variety of use cases Matter addresses – from end nodes to gateways – so device manufacturers can focus on product innovation and accelerating time to market.

This document focuses on NXP’s solution for WiFi enabled platforms using the 88MW32x Microcontroller family.

The Wi-Fi® Microcontroller Platform provides a highly cost-effective, flexible, and easy-to-use hardware/software platform to build a new generation of smart connected devices delivering a broad-range of services to consumers including thermostats, appliances, lighting, home automation and remote access.

The platform is built using our high-performance 88MW32x Cortex-M4F Microcontroller and low power 802.11 b/g/n Wi-Fi® SoCs. It is powered by proven, field-tested EZ-Connect™ Software Development Kit (SDK.) This simplifies development and enables OEMs to focus on value-added features, and delivering applications and services to their customers.

https://www.nxp.com/part/MW320#/
2 HARDWARE SETUP
The minimum hardware required to create and run an end to end Matter setup with i.MX8MM is listed below:

- i.mx8m mini evk - acting as a Thread Border Router and Chip Controller: i.MX8MM-EVKB
- MW320 EVK – acting as Matter Accessory Device: MW320-EVK
3  MATTER ENVIRONMENT SETUP

Matter development relies on open-source resources, leveraging Linux based operating systems like Ubuntu and other tools like git, gcc or python. This also includes GN, a meta build system that generates makefiles and Ninja, a build system meant to replace Make tool.

Matter is available as an open-source SDK containing all the necessary components from scripts to install required tools to stack source code and vendor provided applications.

First step in developing a Matter application is to have Linux support for the build. The recommendation is to have a native Linux machine. If Windows is preferred operating system, support for the build can be set by using:

- Windows Subsystem for Linux (WSL);
- Linux Virtual Machine;

3.1  WSL Ubuntu 20.04 LTS

The Windows Subsystem for Linux (WSL) provides a GNU/Linux environment directly to Windows.

Use the following steps to install the WSL Ubunut 20.04 LTS:

1. On Windows 10, open PowerShell as administrator and run the following commands:
   - Enable the Windows Subsystem for Linux:
     `dism.exe /online /enable-feature /featurename:Microsoft-Windows-Subsystem-Linux /all /norestart`
   - Enable virtual machine feature:
     `dism.exe /online /enable-feature /featurename:VirtualMachinePlatform /all /norestart`
   - Restart your machine to complete the WSL installation
2. Install the Ubuntu 20.04 from Microsoft Store
   - Create a user account and password for your new Linux distribution
Next you will have to check the WSL version and activate WSL 1, if 2 is used. Run PowerShell as administrator:
The following command is used to check the wsl version:
\texttt{wsl --list --verbose}
If wsl version 2 is used, then activate wsl 1 using:
\texttt{wsl --set-version Ubuntu-20.04 1}

After installation is complete, run Ubuntu 20.04 LTS as administrator.

3.2 Linux Virtual Machine
The following steps guide you through virtualbox machine installation steps:

- Download and install virtual machine: \url{https://www.virtualbox.org/wiki/Downloads}
- Please consider that based on your operating system some extra steps will be required for enabling the Virtualization support.
- Download the \texttt{desktop image for Ubuntu 20.04 Focal} and create virtual machine using the downloaded ISO.
- Configure the virtual machine:
  - Linux Ubuntu 20.04 (64 bit);
- VM disk size – more than 20GB;

- Enable USB controller -> USB 1.1(OHCI) Controller (support for USB 2.0 is recommended, if possible, this being available via the Oracle VM VirtualBox Extension Pack);

- Enable network adapter-> Adapter 1 -> Attached to Bridged Adapter;
3.3 Setting up Matter Environment

The following steps guide you through creating Matter build environment.

1. Matter Dependencies:
   - Check for updates and install dos2unix (useful for WSL):

   $ sudo apt update
   $ sudo apt upgrade --y
   $ sudo apt-get install dos2unix

   - Install Matter dependencies:

   $ sudo apt-get install git gcc g++ python pkg-config libssl-dev libdbus-1-dev libglib2.0-dev libavahi-client-dev ninja-build python3-venv python3-dev python3-pip unzip libgirepository1.0-dev libcairo2-dev ninja build python3-venv python3-dev python3-pip unzip libgirepository1.0-dev libcairo2-dev ninja build python3-venv python3-dev python3-pip unzip libgirepository1.0-dev libcairo2-dev

   - Restart the Linux machine.
2. Matter Building Setup instructions:

- Clone the Matter SDK using the public repo:

  $ git clone https://github.com/nxptest/connectedhomeip
  $ cd connectedhomeip
  $ git checkout tags/Ref_kit_v0.92

- Start build environment by running the activate script:
  $ source ./scripts/activate.sh

![Matter Building Setup instructions output]

```
Welcome to...

↓
matter

Boostrap may take a few minutes; please be patient.

Downloading and installing packages into local source directory:
  Setting up CIPD package manager...done (2m18.8s)
  Setting up Python environment...done (1m50.9s)
  Setting up pw packages.............skipped (0.1s)
  Setting up Host tools..............done (0.1s)

Activating environment (setting environment variables):
  Setting environment variables for CIPD package manager...done
  Setting environment variables for Python environment....done
  Setting environment variables for pw packages.............skipped
  Setting environment variables for Host tools..............done

Checking the environment:

2022/10/13 10:25:14 INF Environment passes all checks!
Environment looks good, you are ready to go!
To reactivate this environment in the future, run this in your
terminal:
  source ./activate.sh
To deactivate this environment, run this:
  deactivate
```

![Matter Building Setup instructions output]

4 MW320 MATTER EXAMPLES

In the current Matter SDK for MW320 platform we are providing an implementation for the all-clusters example that was intended to work in different setups as either light node, light switch node or both.

4.1 Matter application building instruction

The following build example will be based on the all-clusters app example:

- Change the directory to the all-clusters application base directory and run the following command in order to download the MW320 SDK as a submodule:

```bash
$ cd examples/all-clusters-app/nxp/mw320
```

```bash
$ git submodule update --init
```

- Generate the build files using GN and start building the application image with the Ninja tool
- The built application can be found in the `/out/debug/` folder

### 4.2 Matter application building instruction

- In order to flash the board the OpenOCD (Open On-Chip Debugger) package

$ sudo apt-get install openocd

- After that the MW320 firmware download image must be prepared:

$ ln -sf third_party/connectedhomeip/third_party/nxp/mw320_sdk/repo mw320_sdk

$ mw320_sdk/tools/mw_img_conv/bin/mw_img_conv mcufw out/debug/chip-mw320-all-clusters-app.bin out/debug/all-cluster-mw320.mcufw.bin 0x1F010000

$ cp out/debug/all-cluster-mw320.mcufw.bin mw320_sdk/mw320_matter_flash/Matter/.
Flashing the firmware image to MW320:

```bash
$ cd mw320_sdk/mw320_matter_flash

$ sudo python2 flashprog.py -l Matter/layout-4m.txt --boot2 Matter/boot2.bin --wififw Matter/mw32x_uapsta_W14.88.36.p172.bin --mcufw Matter/all-cluster-mw320.mcufw.bin -r
```

To catch the application logs a terminal connection needs to be opened to the board:

```bash
$ sudo apt-get install minicom

$ TERM=linux minicom -D /dev/ttyUSB1 -b 115200
```
welcome to mntcon 2.7.1

OPTIONS: ?hRe
Compiled on Dec 23 2019, 02:06:26.
Port /dev/ttyUSB1, 16:44:12

Press CTRL-A Z for help on special keys

*@version: [mw320-2.9.18-005]
+- Init mw320_sdk
call mcuInitPower()
[psn] Error: psn: Could not read key key_size from keystore. Please Flash correct boot2
[PSN]: (start, len)=(0x370000, 0x370000)
MAC Address: 00:50:43:24:13:7e
[net] Initialized TCP/IP networking stack
[mwc29 int complete]
[0.956]CHIP:SH: Initializing CHIP shell commands: 0
[1] mfshav_save_file success
[1] mfshav_save_file success

[6.048]CHIP:DL: Device Configuration:
[6.064]CHIP:DL: Vendor Id: 65521 (0xFFFD)
[6.073]CHIP:DL: Product Id: 12769 (0x4993)
[7.065]CHIP:DL: Setup Discriminator (0xFFFF for UNKNOWN/ERROR): 3840 (0x9f00)
[7.069]CHIP:DL: Manufacturing Date: (not set)
[7.079]CHIP:DL: device Type: 65535 (0xFFFFFFFF)
[7.087]CHIP:SVR: Copy/paste the below URL in a browser to see the QR Code:
[7.096]CHIP:SH: Run CHIP shell Task: 0
5  MATTER NETWORK – TEST SETUP INSTRUCTIONS

The prerequisite step for this part is to have the i.MX8MM as Matter controller. The steps for this setup can be found in “Getting started with i.MX8MM in Matter”.

The following instructions will provide information on how to for a matter test setup containing one controller an one end-nodes: light node

**i.MX8 Mini Matter Controller Setup**

- Connect the I.mx8 mini to the Wi-Fi AP network

```
$modprobe moal mod_para=nxp/wifi_mod_para.conf
$wpa_passphrase ${SSID} ${PASSWORD} > imxrouter.conf
$wpa_supplicant -d -B -i mlan0 -c ./imxrouter.conf
$udhcpc -i mlan0
$echo 1 > /proc/sys/net/ipv6/conf/all/forwarding
$echo 1 > /proc/sys/net/ipv4/ip_forward
$echo 2 > /proc/sys/net/ipv6/conf/all/accept_ra
```

Commission the MW320 board over WiFi

- Connect the MW320 board to the AP

```
>wifi connect SSID PASSWORD
```

- Run the following command to commission the MW320 app over WiFi using the IPv6 global address:

```
$ chip-tool pairing ethernet {NODE_ID_TO_ASSIGN}{SETUP_PIN_CODE}{DISCRIMINATOR} {IPv6_ADDR}{PORT}
```

Where:

- `{NODE_ID_TO_ASSIGN}` - the matter node id;
- `{SETUP_PIN_CODE}` – pin code, default value is 20202021;
- `{DISCRIMINATOR}` – discriminator, default value is 3840;
- `{IPv6_ADDR}` – The global IPv6 address assigned to the MW320 board by the AP’s DHCPv6 server
- `{PORT}` – Web port where the all-clusters app server is opened, default value is 5540
The MW320 all-clusters application is intended to work as either a lighting device or a light-switch device. For this setup we will use it as a light end node.

- You can send On/Off cluster commands to control the GPIO41 LED by the following commands:

To send OnOff cluster ->Toggle command on endpoint 1:

```
$ ./chip-tool onoff toggle {NODE_ID} 1
```

To send OnOff cluster ->On command on endpoint 1:

```
$ ./chip-tool onoff on {NODE_ID} 1
```

To send OnOff cluster ->Off command on endpoint 1:

```
$ ./chip-tool onoff off {NODE_ID} 1
```

To Send Attribute reporting configuration for on-off attribute using min/max interval = 300 seconds, cluster is on the endpoint 1:

```
$ ./chip-tool onoff report on-off 300 301 {NODE_ID} 1
```