TUS SDK 020 Demonstration

Agenda

1. TUS SDK 020 Updates
2. Demonstration
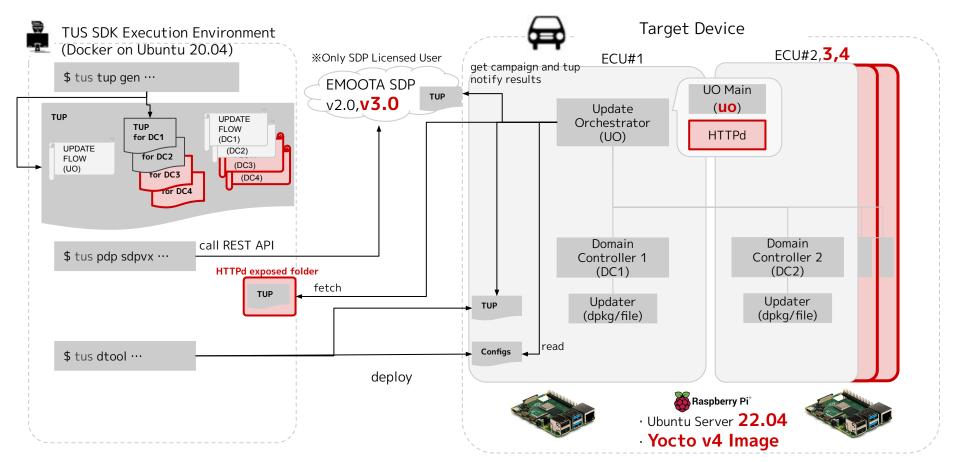
TUS SDK 020 Updates

**TUS SDK Execution Environment
(Docker on Ubuntu 20.04)**

$ tus tup gen …

TUP

TUP
for DC1

for DC2

for DC3

for DC4

UPDATE
FLOW
(UO)

UPDATE
FLOW
(DC1)

(DC2)

(DC3)

(DC4)

$ tus pdp sdpvx …   call REST API

HTTPd exposed folder

TUP   fetch

$ tus dtool …

deploy

※Only SDP Licensed User

**EMOOTA SDP
v2.0, v3.0**

TUP

get campaign and tup
notify results

Target Device

ECU#1   ECU#2, **3,4**

UO Main
(**uo**)

HTTPd

Update
Orchestrator
(UO)

Domain
Controller 1
(DC1)

Domain
Controller 2
(DC2)

Updater
(dpkg/file)

Updater
(dpkg/file)

TUP

Configs   read

Raspberry Pi
・Ubuntu Server **22.04**
・**Yocto v4 Image**

※Red and red frames are updates from TUS SDK 010

TUS SDK 020 Updates Summary

1. TUP Providers
   a. Following TUP Providers are newly supported.
      i. EMOOTA SDP V3.0
      ii. DevHTTPd (TUP can be found on any HTTP server.)
2. Target Device
   a. support Yocto V4 on Raspberrypi
   b. embed HTTP server into UO
   c. add DCs connection management to UO
   d. UO is changed to be a native executable from a Lua script
3. TUP Parser Library
   a. add parsing TUP file from URL
   b. add parallel processing
4. Host Tools
   a. Generator is re-implemented in Native language
   b. generate updateflows for 4 DCs at most

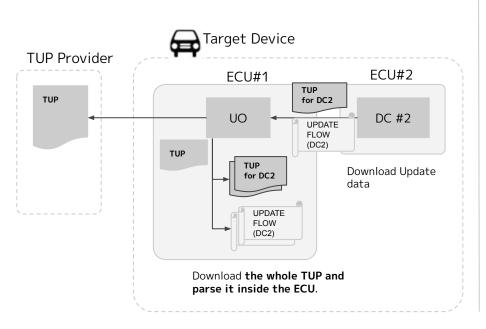※About the details, please refer to the SDK 020 release note.
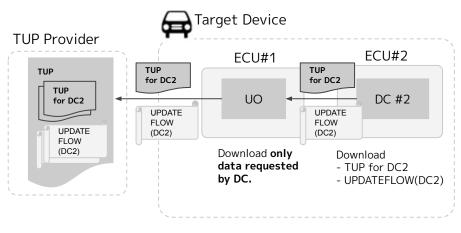
Improve TUP handling on UO

TOYOTA
INFOTECH

By using embedded HTTP and TUP parser's partial download,
UO downloads an only necessary part of TUP as much as needed.

SDK010 TUP handling

TUP Provider

Target Device

ECU#1   ECU#2

TUP
for DC2

UO   DC #2

UPDATE
FLOW
(DC2)

TUP

TUP

TUP
for DC2

UPDATE
FLOW
(DC2)

Download Update
data

Download **the whole TUP and
parse it inside the ECU**.

SDK020 TUP handling

TUP Provider

Target Device

ECU#1   ECU#2

TUP

TUP
for DC2

TUP
for DC2

TUP
for DC2

UO   DC #2

UPDATE
FLOW
(DC2)

UPDATE
FLOW
(DC2)

UPDATE
FLOW
(DC2)

Download **only
data requested
by DC.**

Download
- TUP for DC2
- UPDATEFLOW(DC2)
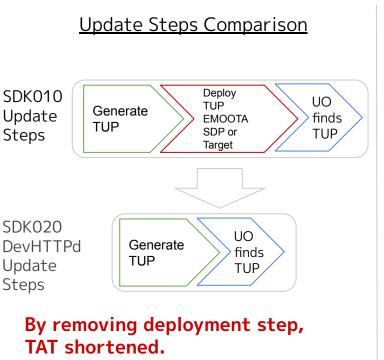
No need to download the whole TUP.

Support DevHTTPd as new TUP Provider

TOYOTA
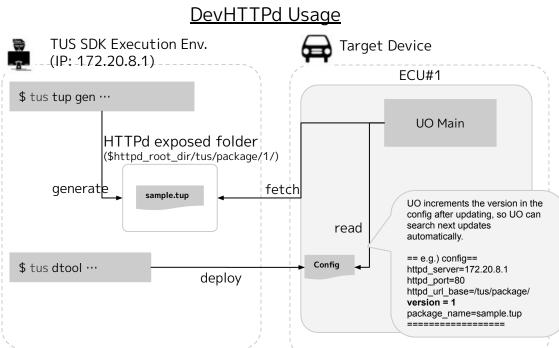**INFOTECH**

Without TUP deployment to Target Device or EMOOTA SDP, UO can find TUP by just placing TUP to HTTP exposed folder.

Update Steps Comparison

SDK010 Update Steps

```
Generate TUP → Deploy TUP EMOOTA SDP or Target → UO finds TUP
```

SDK020 DevHTTPd Update Steps

```
Generate TUP → UO finds TUP
```

By removing deployment step, TAT shortened.

DevHTTPd Usage

TUS SDK Execution Env. (IP: 172.20.8.1)

Target Device

```
$ tus tup gen …
```

HTTPd exposed folder
($httpd_root_dir/tus/package/1/)

generate   sample.tup   fetch

```
$ tus dtool …
```

deploy

ECU#1

UO Main

read

Config

UO increments the version in the config after updating, so UO can search next updates automatically.

```
== e.g.) config==
httpd_server=172.20.8.1
httpd_port=80
httpd_url_base=/tus/package/
version = 1
package_name=sample.tup
==================
```
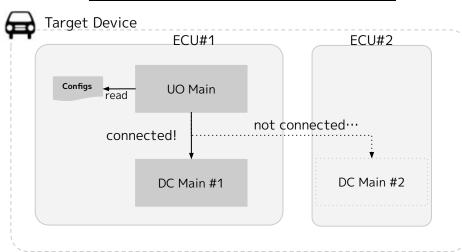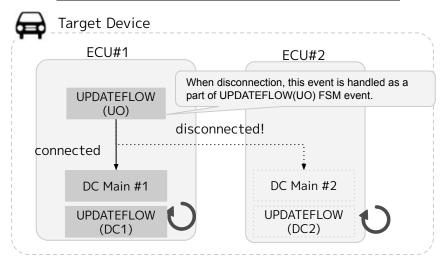
Add DCs connection management to UO

By using this functions,
- At initialization, UO can sync DCs state (e.g. consistency check).
- During the update, UO can detect disconnection from DC.

UO waits for DCs at initialization.

Target Device

ECU#1

ECU#2

Configs ← read — UO Main

connected!

not connected···

DC Main #1

DC Main #2

UO can detect disconnection from DC.

Target Device

ECU#1

ECU#2

UPDATEFLOW (UO)

When disconnection, this event is handled as a part of UPDATEFLOW(UO) FSM event.

connected

disconnected!

DC Main #1

DC Main #2

UPDATEFLOW (DC1)

UPDATEFLOW (DC2)

Demonstration
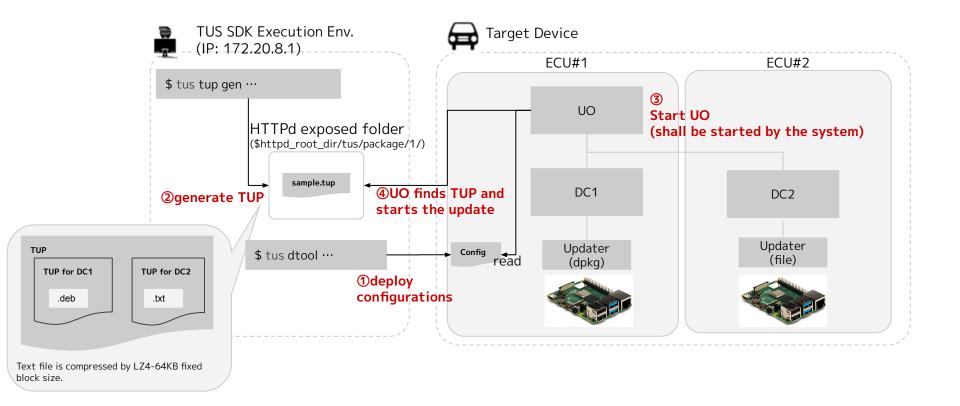
Demo#1 Update by using "DevHTTPd" TUP provider

TUS SDK Execution Env.
(IP: 172.20.8.1)

Target Device

$ tus tup gen …

HTTPd exposed folder
($httpd_root_dir/tus/package/1/)

sample.tup

②generate TUP

④UO finds TUP and
starts the update

$ tus dtool …

TUP

TUP for DC1

.deb

TUP for DC2

.txt

Text file is compressed by LZ4-64KB fixed block size.

①deploy
configurations

ECU#1

ECU#2

UO

③
**Start UO
(shall be started by the system)**

DC1

DC2

Config

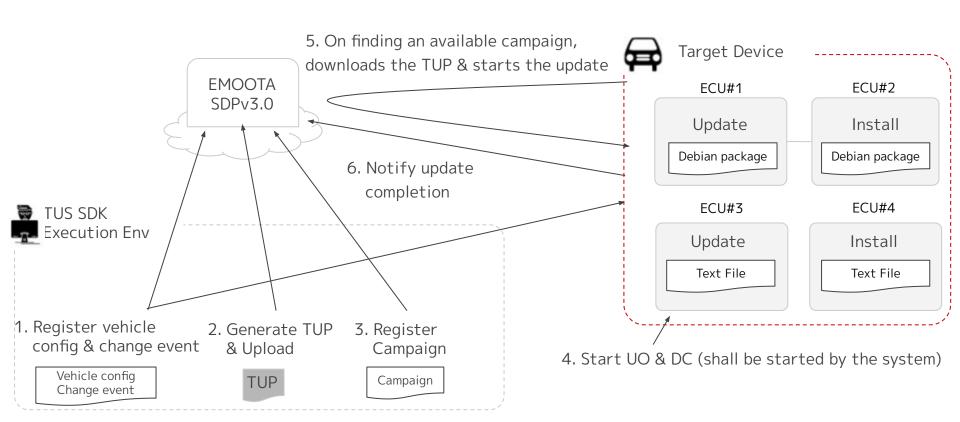read

Updater
(dpkg)

Updater
(file)

Demo#2 Update via OTA Server (SDPv3.0)

*Needs EMOOTA SDP License

5. On finding an available campaign, downloads the TUP & starts the update

EMOOTA SDPv3.0

Target Device

ECU#1
Update
Debian package

ECU#2
Install
Debian package

6. Notify update completion

TUS SDK Execution Env

ECU#3
Update
Text File

ECU#4
Install
Text File

1. Register vehicle config & change event

Vehicle config Change event

2. Generate TUP & Upload

TUP

3. Register Campaign

Campaign

4. Start UO & DC (shall be started by the system)

Appendix.