History				
#	Version	Date	Description	Author
1	1.0	11.29.2021	Hello world OTA example ThingsPro Edge	Amjad Badar
2	1.1	11.30.2021	Added known issues at the end of the document	Amjad Badar

Pre-requisites:
Basic knowledge of Linux, and Microsoft Azure IoT Hub

Upgrade any standard Debian package using Moxa TPE OTA upgrade

In this document we create a simple hello world application in C language and cross compile the binary for target machine architecture e.g. UC computer (armv32). Then we bundle binary into standard .deb package after that we create OTA package using Moxa packer tool and trigger OTA upgrade using Direct Method from Azure IoT Hub.

Development Environment: ubuntu VM 18.04

Steps

1- Create a helloworld.c
2- Cross compile for armhf
3- Test the binary on Target UC-8112A
4- Bundle binary into .deb package
5- Test again on Target UC-8112A (dpkg -i <package_name.deb>)
6- Create OTA URL yaml of the .deb package
7- Upload all OTA files on Azure Blob Storage
8- Trigger OTA upgrade using Direct Method using YAML file

Cross compile for arm32
https://suchprogramming.com/cross-compiling-c-code-for-arm/

```
sudo apt update
sudo apt-get install build-essential
sudo apt install -y build-essential gcc-arm-linux-gnueabihf      # GNU C compiler for armhf architecture
```

Step1

```
ubuntu@ubuntu1804:~/ sudo mkdir helloworld-deb-arm32 && cd helloworld-deb-arm32
ubuntu@ubuntu1804:~/helloworld-deb-arm32$ sudo nano helloworld.c

#include <stdio.h>

int main(void)
{
  printf("Hello World!\nTPE OTA example\n");
  return 0;
}
CTRL+O save and CTRL+X exit

ubuntu@ubuntu1804:~/helloworld-deb-arm32$ ls
helloworld.c
```

Step2:

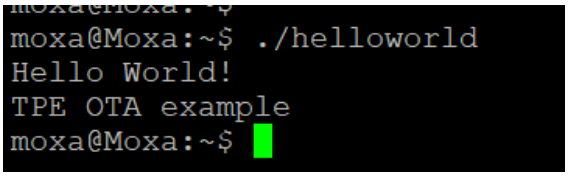
ubuntu@ubuntu1804:~/helloworld-deb-arm32$ sudo arm-linux-gnueabihf-gcc -o helloworld helloworld.c
ubuntu@ubuntu1804:~/helloworld-deb-arm32$ ls
helloworld helloworld.c

Verify binary machine type to make sure it is created for target architecture (Optional)

ubuntu@ubuntu1804:~/helloworld-deb-arm32$ readelf -h helloworld
ELF Header:
 Magic: 7f 45 4c 46 01 01 01 00 00 00 00 00 00 00 00 00
 Class: ELF32
 Data: 2's complement, little endian
 Version: 1 (current)
 OS/ABI: UNIX - System V
 ABI Version: 0
 Type: EXEC (Executable file)
 Machine: ARM

Step:3

copy helloworld binary into UC example /home/moxa/
make it executable sudo chmod 755 helloworld



Step4

https://www.sindastra.de/p/1684/how-to-make-a-basic-debian-and-ubuntu-package-deb-the-easy-way/

Packaging helloworld into .deb

ubuntu@ubuntu1804:~/helloworld-deb-arm32$ sudo mkdir -p helloworld-1.0/DEBIAN
ubuntu@ubuntu1804:~/helloworld-deb-arm32$ sudo mkdir -p helloworld-1.0/usr/local/bin
ubuntu@ubuntu1804:~/helloworld-deb-arm32$ sudo touch helloworld-1.0/DEBIAN/control
ubuntu@ubuntu1804:~/helloworld-deb-arm32$ sudo mv helloworld helloworld-1.0/usr/local/bin/

Create meta data control file example

ubuntu@ubuntu1804:~/helloworld-deb-arm32$ sudo nano helloworld-1.0/DEBIAN/control

Package: helloworld
Version: 1.0
Architecture: all
Maintainer: Amjad <noreply@localhost>
Depends: libc6
Installed-Size: 20
Homepage: https://github.com/sindastra
Description: Example helloworld
 This is an example application named helloworld for a Debian packaging tutorial.

CTRL+O save and CTRL+X exit

ubuntu@ubuntu1804:~/helloworld-deb-arm32$ sudo chown -R 0:0 helloworld-1.0/
ubuntu@ubuntu1804:~/helloworld-deb-arm32$ sudo dpkg -b helloworld-1.0/

bundle in tar.gz for export into UC

ubuntu@ubuntu1804:~/helloworld-deb-arm32$ cd ..
ubuntu@ubuntu1804:~$ sudo tar -czvf helloworld-deb-arm32.tar.gz helloworld-deb-arm32/

Step5

Download helloworld-deb-arm32.tar.gz into UC and extract the archive

Unzip command

tar xvf helloworld-deb-arm32.tar.gz

moxa@Moxa:~$ cd helloworld-deb-arm32
moxa@Moxa:~/helloworld-deb-arm32$ ls
helloworld-1.0 helloworld-1.0.deb helloworld.c

moxa@Moxa:~/helloworld-deb-arm32$ sudo dpkg -i helloworld-1.0.deb
(Reading database ... 30731 files and directories currently installed.)
Preparing to unpack helloworld-1.0.deb ...
Unpacking helloworld (1.0) over (1.0) ...
Setting up helloworld (1.0) ...
moxa@Moxa:~/helloworld-deb-arm32$ helloworld
Hello World!
TPE OTA example

list installed package

moxa@Moxa:~/helloworld-deb-arm32$ apt list --installed | grep helloworld

Remove

moxa@Moxa:~/helloworld-deb-arm32$ sudo apt remove helloworld

Once everything works you should change the path from /usr/local to /usr so it doesn't clash with "local" packages

Step6

https://moxaeuiiot.blob.core.windows.net/ota-upgrade/ThingsPro%20Edge%20OTA%20Upgrade%20Pack%20Guide%20.html

bundle deb package for OTA package using TPE packer tool

ubuntu@ubuntu1804:~$ sudo docker pull moxa2019/thingspro-upgrade-packer:latest
ubuntu@ubuntu1804:~$ sudo mkdir packer

ubuntu@ubuntu1804:~$ sudo su
root@ubuntu1804:/home/ubuntu# sudo docker run -it --rm -u ${UID} -v pwd:/data moxa2019/thingspro-upgrade-packer create

OUTPUT
./
package.yaml
        352 92%   0.00kB/s   0:00:00 (xfr#1, to-chk=6/8)
data/
data/.keep
        352 92%   0.00kB/s   0:00:00 (xfr#2, to-chk=4/8)
data/task1/
data/task1/install
        382 100%   29.30kB/s   0:00:00 (xfr#3, to-chk=2/8)
data/task1/debs/
data/task1/debs/.keep
        382 100%   29.30kB/s   0:00:00 (xfr#4, to-chk=0/8)
Completed

root@ubuntu1804:/home/ubuntu# cd packer/
root@ubuntu1804:/home/ubuntu/packer# ls
data package.yaml

root@ubuntu1804:/home/ubuntu/packer# sudo nano package.yaml

```
kind: package
version: v1
metadata:
 name: update-helloworld
 version: 1.0
 arch: armhf
spec:
 location: to-be-filled-by-tool
 packages:
  - name: update helloworld debian package
    displayName: Install Helloworld Debian Package
    path: helloworld-1.0
    version: 0.1
  # - name: task2
  #   displayName: Install MPKG
  #   path: edge-web_*_armhf.mpkg
  #   version: to-be-filled-by-tool
```

CTRL+O save and CTRL+X exit

```
ubuntu@ubuntu1804:~/packer$ cd ..
root@ubuntu1804:/home/ubuntu# mkdir packer/data/helloworld-1.0
root@ubuntu1804:/home/ubuntu# mkdir packer/data/helloworld-1.0/debs

ubuntu@ubuntu1804:~$ cd helloworld-deb-arm32/
ubuntu@ubuntu1804:~/helloworld-deb-arm32$ cp helloworld-1.0.deb /home/ubuntu/packer/data/helloworld-1.0/debs
ubuntu@ubuntu1804:~/helloworld-deb-arm32$ sudo nano /home/ubuntu/packer/data/helloworld-1.0/install
```

```
#!/bin/sh

dpkg -i debs/*.deb
```

CTRL+O save and CTRL+X exit

Make sure install file is executable

```
ubuntu@ubuntu1804:~/helloworld-deb-arm32$ cd /home/ubuntu/packer/
ubuntu@ubuntu1804:~/packer$ sudo su
root@ubuntu1804:/home/ubuntu/packer# sudo docker run -it --rm -u ${UID} -v `pwd`:/data moxa2019/thingspro-upgrade-packer pack

total 5.2M
-rw-r--r-- 1 root root 5.2M Nov 28 16:03 update-helloworld_1.0_armhf.deb
-rw-r--r-- 1 root root  360 Nov 28 16:03 update-helloworld_1.0_armhf.deb.zsync
-rw-r--r-- 1 root root  278 Nov 28 16:03 update-helloworld_1.0_armhf.yaml
Completed

root@ubuntu1804:/home/ubuntu/packer# ls
build  data  package.yaml

root@ubuntu1804:/home/ubuntu/packer# cd build/
root@ubuntu1804:/home/ubuntu/packer/build# ls
armhf

root@ubuntu1804:/home/ubuntu/packer/build# cd armhf/
root@ubuntu1804:/home/ubuntu/packer/build/armhf# ls
update-helloworld_1.0_armhf.deb  update-helloworld_1.0_armhf.deb.zsync  update-helloworld_1.0_armhf.yaml
```

Bundle into tar.gz for export

```
root@ubuntu1804:/home/ubuntu/packer/build/armhf# cd ..
root@ubuntu1804:/home/ubuntu/packer/build# sudo tar -czvf armhf.tar.gz armhf/
```

Step7

Unzip tar.gz and import files in downloading server example Azure blob storage

All 3 files (.deb, .zsync, .yaml) uploaded to same place of web server, for example https://abc.com/sw/
When you give TPE yaml url, https://abc.com/sw/update-helloworld_1.0_armhf.yaml

TPE will download the other two file from same URL path

Webserver example: Azure Storage OTA URL
https://moxaeuiiot.blob.core.windows.net/ota-upgrade/update-helloworld_1.0_armhf.yaml

Step8

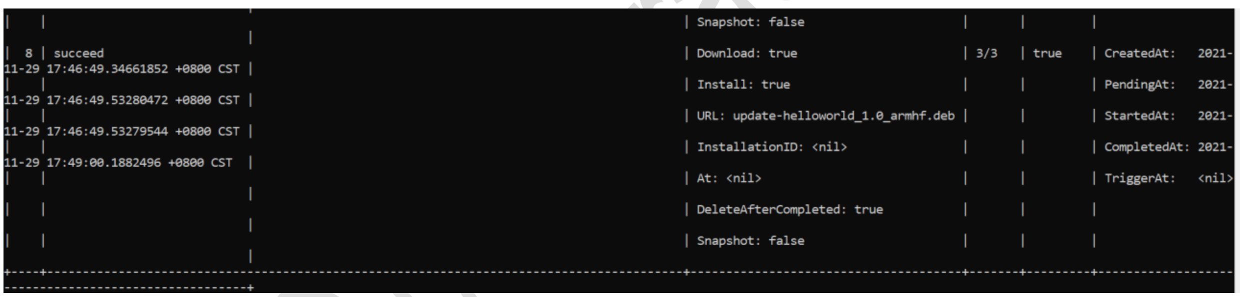
Invoke Direct Method ThingsPro API
https://thingspro-edge.moxa.online/v2.2.0/thingspro-agent/index.html

Method Name:
thingspro-software-upgrade

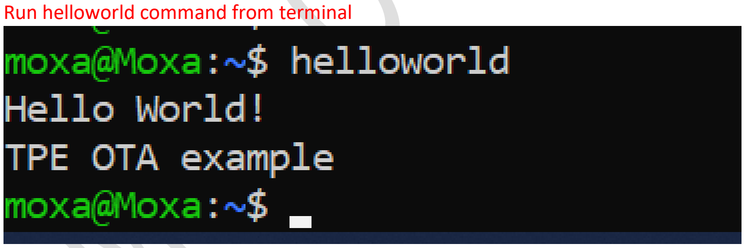
Payload:

```
{
  "downloadURL": "https://moxaeuiiot.blob.core.windows.net/ota-upgrade/update-helloworld_1.0_armhf.yaml",
  "runInstallation": true
}
```

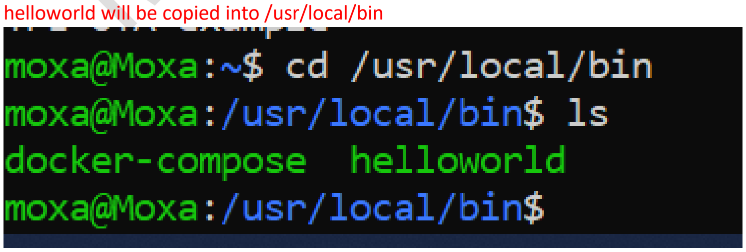
Check upgrade status on UC
sudo appman upgrade ls


| | | Snapshot: false | | |
| 8 | succeed | Download: true | 3/3 | true | CreatedAt: 2021-
11-29 17:46:49.34661852 +0800 CST |
| | | Install: true | | | PendingAt: 2021-
11-29 17:46:49.53280472 +0800 CST |
| | | URL: update-helloworld_1.0_armhf.deb | | | StartedAt: 2021-
11-29 17:46:49.53279544 +0800 CST |
| | | InstallationID: <nil> | | | CompletedAt: 2021-
11-29 17:49:00.1882496 +0800 CST |
| | | At: <nil> | | | TriggerAt: <nil>
| | | DeleteAfterCompleted: true | | |
| | | Snapshot: false | | |

Run helloworld command from terminal



helloworld will be copied into /usr/local/bin



ThingsPro agent will send back upgrade progress and status via reported properties. See thingspro-agent module TWIN in case of Azure IoT Edge and device TWIN in case for Azure IoT Device

Known ISSUES:

At **Step6** in PDF when generating the OTA packages, there is known issue packer tool doesn't update the location/name of the package in the YAML file.

root@ubuntu1804:/home/ubuntu/packer# sudo docker run -it --rm -u ${UID} -v `pwd`:/data moxa2019/thingspro-upgrade-packer pack

total 5.2M
-rw-r--r-- 1 root root 5.2M Nov 28 16:03 update-helloworld_1.0_armhf.deb
-rw-r--r-- 1 root root 360 Nov 28 16:03 update-helloworld_1.0_armhf.deb.zsync
-rw-r--r-- 1 root root 278 Nov 28 16:03 update-helloworld_1.0_armhf.yaml
Completed

Please use exactly same name in YAML file (update-helloworld_1.0_armhf.deb) as it generated by above command

update-helloworld_1.0_armhf.yaml

```yaml
kind: package
version: v1
metadata:
  arch: armhf
  name: update-helloworld
  version: 1
spec:
  location: update-helloworld 1.0 armhf.deb
  packages:
  - name: update helloworld debian package
    displayName: Install Helloworld Debian Package
    path: helloworld-1.0
    version: "0.1"
```