## Custom MQTT Application

### Description

Custom MQTT (custom\_mqtt) application is used to demonstrate MQTT client on Talaria TWO.

This application can download the certificates onto Talaria TWO, start the MQTT connection, subscribe to a topic, publish message to MQTT server and close the MQTT connection.

This application uses MQTT message IDs and group IDs provided by InnoPhase IoT as well as custom defined IDs.

### Prerequisites

1. GTKTerm or similar application.
2. Access Point configured with WPA/WPA2 personal security.
3. Mosquitto.exe from <http://mosquitto.org/download/>.
4. Generate certificates from:
   1. Download certificate authority file: <https://test.mosquitto.org/>.
   2. Generate client certificates: <https://test.mosquitto.org/ssl/>.

**Note**: Ensure Talaria TWO is connected to an Access Point to run the custom MQTT application.

### Command Description

Connect to an Access Point with specified SSID and passphrase.

|  |
| --- |
| $ ./conmgr connect <SSID> <passphrase> |

Store the certificate.

|  |
| --- |
| ./custom\_mqtt {certstore} <filename> <store as> |

where,

1. filename: Certificate file path (absolute) as on the Host
2. store as: Path to be stored as in Talaria TWO filesystem. Must start with */data/*

Start MQTT connection with Mosquitto server.

|  |
| --- |
| $ ./custom\_mqtt {mqttinit} <server name> <port number> <transport> <mqtt id> <usr name> <password> [ca cert] [client cert] [client key] |

where,

1. server name: MQTT broker address
2. port number: Port number
3. transport: Transport mode 0 -> TCP , 1-> TLS with Client auth
4. mqtt id: MQTT client ID
5. usr name: Username (use --no\_un if no username is required)
6. password: Password (use --no\_pw if no password is required)
7. ca cert: CA certificate name
8. client cert: Client certificate name
9. client key: Client key certificate name

Close MQTT connection.

|  |
| --- |
| $ ./custom\_mqtt {mqttclose} |

### Procedure

Execute the following operations on the Talaria TWO:

**Step 1**: Connect Talaria TWO to the desired network by providing SSID and passphrase.

**Step 2**: Store the client certificates on Talaria TWO.

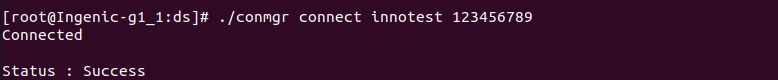
**Step 3**: Start MQTT connection with Mosquitto server.

**Step 4**: Connect the backend Host machine to the same network as Talaria TWO and issue the commands to start MQTT client on Linux terminal to observe publish/subscribe messages.

**Step 5**: Close the MQTT connection.

### Expected Output

#### Host Console Logs



A screenshot of a computer program

Description automatically generated

A computer screen with white text

Description automatically generated

A computer screen shot

Description automatically generated

Figure 3: custom\_mqtt - Host console logs

|  |
| --- |
| [root@:ds]# reboot  [root@:ds]# umount: tmpfs busy - remounted read-only  The system is going down NOW!  Sent SIGTERM to all processes  Sent SIGKILL to all processes  Requesting system reboot[ 147.951504] sw\_read: ERESTARTSYS  [ 147.959702] sw\_release  [ 147.962204] sw\_fasync: fd=-1, file=807a5960, mode=0  [ 148.951328] Restarting system.  [ 148.954522] Restarting after 4 ms    U-Boot SPL 2013.07 (Nov 14 2022 - 10:37:43)  Timer init  CLK stop  PLL init  pll\_init:366  pll\_cfg.pdiv = 10, pll\_cfg.h2div = 5, pll\_cfg.h0div = 5, pll\_cfg.cdiv = 1, pll\_2  nf=116 nr = 1 od0 = 1 od1 = 2  cppcr is 07405100  CPM\_CPAPCR 0740510d  nf=100 nr = 1 od0 = 1 od1 = 2  cppcr is 06405100  CPM\_CPMPCR 0640510d  nf=100 nr = 1 od0 = 1 od1 = 2  cppcr is 06405100  CPM\_CPVPCR 0640510d  cppcr 0x9a7b5510  apll\_freq 1392000000  mpll\_freq 1200000000  vpll\_freq = 1200000000  ddr sel mpll, cpu sel apll  ddrfreq 600000000  cclk 1392000000  l2clk 696000000  h0clk 240000000  h2clk 240000000  pclk 120000000  CLK init  SDRAM init  sdram init start  ddr\_inno\_phy\_init ..!  phy reg = 0x00000007, CL = 0x00000007  ddr\_inno\_phy\_init ..! 11: 00000004  ddr\_inno\_phy\_init ..! 22: 00000006  ddr\_inno\_phy\_init ..! 33: 00000006  REG\_DDR\_LMR: 00000210  REG\_DDR\_LMR: 00000310  REG\_DDR\_LMR: 00000110  REG\_DDR\_LMR, MR0: 00f73011  T31\_0x5: 00000007  T31\_0x15: 0000000c  T31\_0x4: 00000000  T31\_0x14: 00000002  INNO\_TRAINING\_CTRL 1: 00000000  INNO\_TRAINING\_CTRL 2: 000000a1  T31\_cc: 00000003  INNO\_TRAINING\_CTRL 3: 000000a0  T31\_118: 0000003c  T31\_158: 0000003c  T31\_190: 0000001e  T31\_194: 0000001c  jz-04 : 0x00000051  jz-08 : 0x000000a0  jz-28 : 0x00000024  DDR PHY init OK  INNO\_DQ\_WIDTH :00000003  INNO\_PLL\_FBDIV :00000014  INNO\_PLL\_PDIV :00000005  INNO\_MEM\_CFG :00000051  INNO\_PLL\_CTRL :00000018  INNO\_CHANNEL\_EN :0000000d  INNO\_CWL :00000006  INNO\_CL :00000007  DDR Controller init  DDRC\_STATUS 0x80000001  DDRC\_CFG 0x0aa88a42  DDRC\_CTRL 0x0000011c  DDRC\_LMR 0x00400008  DDRC\_DLP 0x00000000  DDRC\_TIMING1 0x050f0a06  DDRC\_TIMING2 0x021c0a07  DDRC\_TIMING3 0x200a0722  DDRC\_TIMING4 0x26240031  DDRC\_TIMING5 0xff060405  DDRC\_TIMING6 0x321c0505  DDRC\_REFCNT 0x00910603  DDRC\_MMAP0 0x000020f8  DDRC\_MMAP1 0x00002800  DDRC\_REMAP1 0x030e0d0c  DDRC\_REMAP2 0x07060504  DDRC\_REMAP3 0x0b0a0908  DDRC\_REMAP4 0x0f020100  DDRC\_REMAP5 0x13121110  DDRC\_AUTOSR\_EN 0x00000000  sdram init finished  SDRAM init ok  board\_init\_r  image entry point: 0x80100000      U-Boot 2013.07 (Nov 14 2022 - 10:37:43)    Board: ISVP (XXXXXX)  DRAM: 128 MiB  Top of RAM usable for U-Boot at: 84000000  Reserving 443k for U-Boot at: 83f90000  Reserving 32784k for malloc() at: 81f8c000  Reserving 32 Bytes for Board Info at: 81f8bfe0  Reserving 124 Bytes for Global Data at: 81f8bf64  Reserving 128k for boot params() at: 81f6bf64  Stack Pointer at: 81f6bf48  Now running in RAM - U-Boot at: 83f90000  MMC: msc: 0  the manufacturer c8  SF: Detected GD25Q128    In: serial  Out: serial  Err: serial  Net: ====>PHY not found!Jz4775-9161  Hit any key to stop autoboot: 0  the manufacturer c8  SF: Detected GD25Q128    --->probe spend 4 ms  SF: 2621440 bytes @ 0x40000 Read: OK  --->read spend 843 ms  ## Booting kernel from Legacy Image at 80600000 ...  Image Name: Linux-3.10.14\_\_isvp\_swan\_1.0\_\_  Image Type: MIPS Linux Kernel Image (lzma compressed)  Data Size: 1994978 Bytes = 1.9 MiB  Load Address: 80010000  Entry Point: 80450890  Verifying Checksum ... OK  Uncompressing Kernel Image ... OK    Starting kernel ...    [ 0.000000] Initializing cgroup subsys cpu  [ 0.000000] Initializing cgroup subsys cpuacct  [ 0.000000] Linux version 3.10.14\_\_isvp\_swan\_1.0\_\_ (lesly@synergic) (gcc ver3  [ 0.000000] bootconsole [early0] enabled  [ 0.000000] CPU0 RESET ERROR PC:8020DE80  [ 0.000000] [<8020de80>] \_\_delay+0x0/0x10  [ 0.000000] CPU0 revision is: 00d00100 (XXXXX)  [ 0.000000] FPU revision is: 00b70000  [ 0.000000] CCLK:1392MHz L2CLK:696Mhz H0CLK:200MHz H2CLK:200Mhz PCLK:100Mhz  [ 0.000000] Determined physical RAM map:  [ 0.000000] memory: 00593000 @ 00010000 (usable)  [ 0.000000] memory: 0003d000 @ 005a3000 (usable after init)  [ 0.000000] User-defined physical RAM map:  [ 0.000000] memory: 04000000 @ 00000000 (usable)  [ 0.000000] Zone ranges:  [ 0.000000] Normal [mem 0x00000000-0x03ffffff]  [ 0.000000] Movable zone start for each node  [ 0.000000] Early memory node ranges  [ 0.000000] node 0: [mem 0x00000000-0x03ffffff]  [ 0.000000] Primary instruction cache 32kB, 8-way, VIPT, linesize 32 bytes.  [ 0.000000] Primary data cache 32kB, 8-way, VIPT, no aliases, linesize 32 bys  [ 0.000000] pls check processor\_id[0x00d00100],sc\_jz not support!  [ 0.000000] MIPS secondary cache 128kB, 8-way, linesize 32 bytes.  [ 0.000000] Built 1 zonelists in Zone order, mobility grouping off. Total p6  [ 0.000000] Kernel command line: console=ttyS1,115200n8 mem=64M@0x0 rmem=64M)  [ 0.000000] PID hash table entries: 256 (order: -2, 1024 bytes)  [ 0.000000] Dentry cache hash table entries: 8192 (order: 3, 32768 bytes)  [ 0.000000] Inode-cache hash table entries: 4096 (order: 2, 16384 bytes)  [ 0.000000] Memory: 58512k/65536k available (4389k kernel code, 7024k reserv)  [ 0.000000] SLUB: HWalign=32, Order=0-3, MinObjects=0, CPUs=1, Nodes=1  [ 0.000000] Preemptible hierarchical RCU implementation.  [ 0.000000] NR\_IRQS:358  [ 0.000000] clockevents\_config\_and\_register success.  [ 0.000040] Calibrating delay loop... 1386.49 BogoMIPS (lpj=2772992)  [ 0.029794] pid\_max: default: 32768 minimum: 301  [ 0.034662] Mount-cache hash table entries: 512  [ 0.039568] Initializing cgroup subsys debug  [ 0.043856] Initializing cgroup subsys freezer  [ 0.049935] regulator-dummy: no parameters  [ 0.054191] NET: Registered protocol family 16  [ 0.069542] bio: create slab <bio-0> at 0  [ 0.075026] jz-dma jz-dma: JZ SoC DMA initialized  [ 0.080068] SCSI subsystem initialized  [ 0.083958] usbcore: registered new interface driver usbfs  [ 0.089492] usbcore: registered new interface driver hub  [ 0.094928] usbcore: registered new device driver usb  [ 0.100124] i2c-gpio i2c-gpio.1: using pins 57 (SDA) and 58 (SCL)  [ 0.106312] (null): set:249 hold:250 dev=100000000 h=500 l=500  [ 0.113771] Switching to clocksource jz\_clocksource  [ 0.118697] cfg80211: Calling CRDA to update world regulatory domain  [ 0.125578] NET: Registered protocol family 2  [ 0.130351] TCP established hash table entries: 512 (order: 0, 4096 bytes)  [ 0.137294] TCP bind hash table entries: 512 (order: -1, 2048 bytes)  [ 0.143710] TCP: Hash tables configured (established 512 bind 512)  [ 0.150011] TCP: reno registered  [ 0.153251] UDP hash table entries: 256 (order: 0, 4096 bytes)  [ 0.159166] UDP-Lite hash table entries: 256 (order: 0, 4096 bytes)  [ 0.165703] NET: Registered protocol family 1  [ 0.170302] RPC: Registered named UNIX socket transport module.  [ 0.176280] RPC: Registered udp transport module.  [ 0.181008] RPC: Registered tcp transport module.  [ 0.185774] RPC: Registered tcp NFSv4.1 backchannel transport module.  [ 0.192594] freq\_udelay\_jiffys[0].max\_num = 10  [ 0.197048] cpufreq udelay loops\_per\_jiffy  [ 0.201442] 12000 23905 23905  [ 0.204699] 24000 47810 47810  [ 0.207973] 60000 119525 119525  [ 0.211416] 120000 239051 239051  [ 0.214948] 200000 398418 398418  [ 0.218479] 300000 597627 597627  [ 0.222010] 600000 1195255 1195255  [ 0.225719] 792000 1577736 1577736  [ 0.229427] 1008000 2008028 2008028  [ 0.233224] 1200000 2390510 2390510  [ 0.241282] squashfs: version 4.0 (2009/01/31) Phillip Lougher  [ 0.247798] jffs2: version 2.2. © 2001-2006 Red Hat, Inc.  [ 0.253627] msgmni has been set to 114  [ 0.258430] io scheduler noop registered  [ 0.262430] io scheduler cfq registered (default)  [ 0.268212] jz-uart.1: ttyS1 at MMIO 0x10031000 (irq = 58) is a uart1  [ 0.275802] console [ttyS1] enabled, bootconsole disabled  [ 0.275802] console [ttyS1] enabled, bootconsole disabled  [ 0.289612] brd: module loaded  [ 0.294168] loop: module loaded  [ 0.297913] zram: Created 2 device(s) ...  [ 0.302145] logger: created 256K log 'log\_main'  [ 0.307206] jz TCU driver register completed  [ 0.311969] wait stable.[289][cgu\_ssi]  [ 0.315926] the id code = c84018, the flash name is GD25Q127C  [ 0.321864] the flash->board\_info->quad\_mode = 6b  [ 0.326718] JZ SFC Controller for SFC channel 0 driver register  [ 0.332842] 4 cmdlinepart partitions found on MTD device jz\_sfc  [ 0.338974] Creating 4 MTD partitions on "jz\_sfc":  [ 0.343927] 0x000000000000-0x000000040000 : "boot"  [ 0.349298] 0x000000040000-0x0000002c0000 : "kernel"  [ 0.354790] 0x0000002c0000-0x000000ac0000 : "root"  [ 0.360112] 0x000000ac0000-0x000001000000 : "appfs"  [ 0.365522] SPI NOR MTD LOAD OK  [ 0.368857] dma dma0chan17: Channel 17 have been requested.(phy id 14,type 0)  [ 0.377805] dma dma0chan18: Channel 18 have been requested.(phy id 13,type 0)  [ 0.386874] jz-ssi jz-ssi.0: master is unqueued, this is deprecated  [ 0.393387] JZ SSI Controller for SPI channel 0 driver register  [ 0.399546] tun: Universal TUN/TAP device driver, 1.6  [ 0.404770] tun: (C) 1999-2004 Max Krasnyansky <maxk@qualcomm.com>  [ 0.411236] usbcore: registered new interface driver zd1201  [ 0.417073] jz-dwc2 jz-dwc2: cgu clk gate get error  [ 0.422144] DWC IN OTG MODE  [ 0.425658] dwc2 dwc2: Keep PHY ON  [ 0.429174] dwc2 dwc2: Using Buffer DMA mode  [ 0.433588] dwc2 dwc2: Core Release: 3.00a  [ 0.437854] dwc2 dwc2: DesignWare USB2.0 High-Speed Host Controller  [ 0.444340] dwc2 dwc2: new USB bus registered, assigned bus number 1  [ 0.451532] hub 1-0:1.0: USB hub found  [ 0.455444] hub 1-0:1.0: 1 port detected  [ 0.459602] dwc2 dwc2: DWC2 Host Initialized  [ 0.464180] usbcore: registered new interface driver usb-storage  [ 0.470642] jzmmc\_v1.2 jzmmc\_v1.2.0: vmmc regulator missing  [ 0.476652] jzmmc\_v1.2 jzmmc\_v1.2.0: register success!  [ 0.482076] jzmmc\_v1.2 jzmmc\_v1.2.1: vmmc regulator missing  [ 0.487959] jzmmc\_v1.2 jzmmc\_v1.2.1: register success!  [root@:~]# cd config/ds  [root@:ds]# ./start.sh  [root@:ds]# [ 36.834768] wait stable.[289][cgu\_msc1]  [ 36.840434] cs:0x7, vnd:0xABCD, dev:0x1  [ 36.844401] mmc1: new SDIO card at address 0001  [ 36.851589] sdio\_t2\_probe: 793  [ 36.857290] sw\_uevent  [ 36.861765] New device wlanSDIO0  [root@:ds]# ./tunadapter board\_conf=board.conf&  [root@:ds]# board\_conf file=0  Key file path = /etc  Loading conf from board.conf  Config Loaded: board.conf  =========conf========  platform=t31\_without\_powersave  host\_irq\_num=12  host\_irq\_dev=  t2\_irq\_pin=0  t2\_irq\_mode=0  shutdown\_gpio\_s[ 42.685728] sw\_open  tates=pppppp--------p--p--p-  powersave\_support=0  wakeup\_pin=39  t2\_wakeup\_pin=14  wakeup\_level\_at\_t2=1  delay1=0  delay2=0  sleep\_udelay=500  =====================  ./tunadapter: platform=t31\_without\_powersave  Initializing hapi  Opening Serial device /dev/wlanSDIO0  Speed=0    hapi\_recv\_thread:992 DEBUG:recv thread entry  hapi\_config:1606 DEBUG:hapi\_config. 0 0 0 0  hapi\_config:1666 DEBUG:before calling hio\_query  hapi\_hio\_query:396 DEBUG:Connected to T2, max packet size is 4092  hapi\_hio\_query:397 DEBUG:Firmware version: , patch version  hapi\_hio\_query:398 DEBUG:Hapi version: 1  hapi\_config:1668 DEBUG:after calling hio\_query  Initializing tunadaptor interface  hapi\_wcm\_create:112 DEBUG:WiFi interface created with mac addr: E2:69:3A:00:04:E  Tunadaptor ready...  ptsname:/dev/pts/0    [root@:ds]# ./conmgr connect innotest 123456789  Connected    Status : Success  [root@:ds]# ./custom\_mqtt certstore aws\_root\_ca.crt /data/aws\_root\_c  a.crt  action:certstore:4  Dest cert name(path on t2) /data/aws\_root\_ca.crt    Send len = 256, index = 0 size=1188  Send len = 256, index = 256 size=932  Send len = 256, index = 512 size=676  Send len = 256, index = 768 size=420  Send len = 164, index = 1024 size=164[root@:ds]#  [root@:ds]#  [root@:ds]#  [root@:ds]# ./custom\_mqtt certstore aws\_device\_cert.crt /data/aws\_de  vice\_cert.crt  action:certstore:4  Dest cert name(path on t2) /data/aws\_device\_cert.crt    Send len = 256, index = 0 size=1225  Send len = 256, index = 256 size=969  Send len = 256, index = 512 size=713  Send len = 256, index = 768 size=457  Send len = 201, index = 1024 size=201[root@:ds]#  [root@:ds]#  [root@:ds]# ./custom\_mqtt certstore aws\_device\_pkey.key /data/aws\_de  vice\_pkey.key  action:certstore:4  Dest cert name(path on t2) /data/aws\_device\_pkey.key    Send len = 256, index = 0 size=1676  Send len = 256, index = 256 size=1420  Send len = 256, index = 512 size=1164  Send len = 256, index = 768 size=908  Send len = 256, index = 1024 size=652  Send len = 256, index = 1280 size=396  Send len = 140, index = 1536 size=140[root@:ds]#  [root@:ds]#  [root@ ds]# ./custom\_mqtt mqttinit a3t0o11ohwlo2h-ats.iot.ap-south-1  .amazonaws.com 8883 1 T2\_MAC apps-dev@innophaseiot.com InnoCloud2020 /data/aws\_r  oot\_ca.crt /data/aws\_device\_cert.crt /data/aws\_device\_pkey.key  action:mqttinit:11    MQTT client connect status = 0    MQTT client connect Success. mqtt\_handle = 707144 (aca48)  mqtt\_publish 95    Status : Success  [root@:ds]# ./custom\_mqtt mqttclose  action:mqttclose:2    MQTT client disconnect sussess. status = 0  [root@:ds]# |

### List of Message IDs Used

This application makes use of following message IDs and uses Group numbers 68 (HIO\_GROUP\_MQTT\_CUSTOM), 80 (HIO\_GROUP\_FILE), 71(HIO\_GROUP\_MQTT):

1. FILE\_HIO\_ADD\_REQ

This message will be sent to Talaria TWO when this application is invoked to store certificate files in Talaria TWO. Path of the file to be stored as, in the filesystem will be sent to Talaria TWO prior to certificate DATA to Talaria TWO.

1. FILE\_HIO\_SEND\_REQ

This message is sent to Talaria TWO to store the certificate. The certificate file is sent in multiple chunks of bytes. Hence, this message is sent with each chunk. Talaria TWO receives this message and replies with FILE\_HIO\_SEND\_RSP with status.

1. FILE\_HIO\_DELETE\_REQ

This message will be sent to Talaria TWO to remove the certificate files from Talaria TWO filesystem when invoked with “certdel”

1. MQTT\_CLIENT\_CONNECT\_REQ

This message is sent to Talaria TWO to start the MQTT client. Talaria TWO receives this message and replies with MQTT\_CLIENT\_CONNECT\_RSP along with the status.

1. MQTT\_CLIENT\_DISCONNECT\_REQ

This message is sent to Talaria TWO to close the MQTT client. Talaria TWO receives this message and replies with MQTT\_CLIENT\_DISCONNECT\_RSP along with the status.

1. MQTT\_CUSTOM\_ITERATIVE\_PUBLISH\_REQ

This message will be sent to Talaria TWO to request for publish message from MQTT server when there is any publish message received on the server.