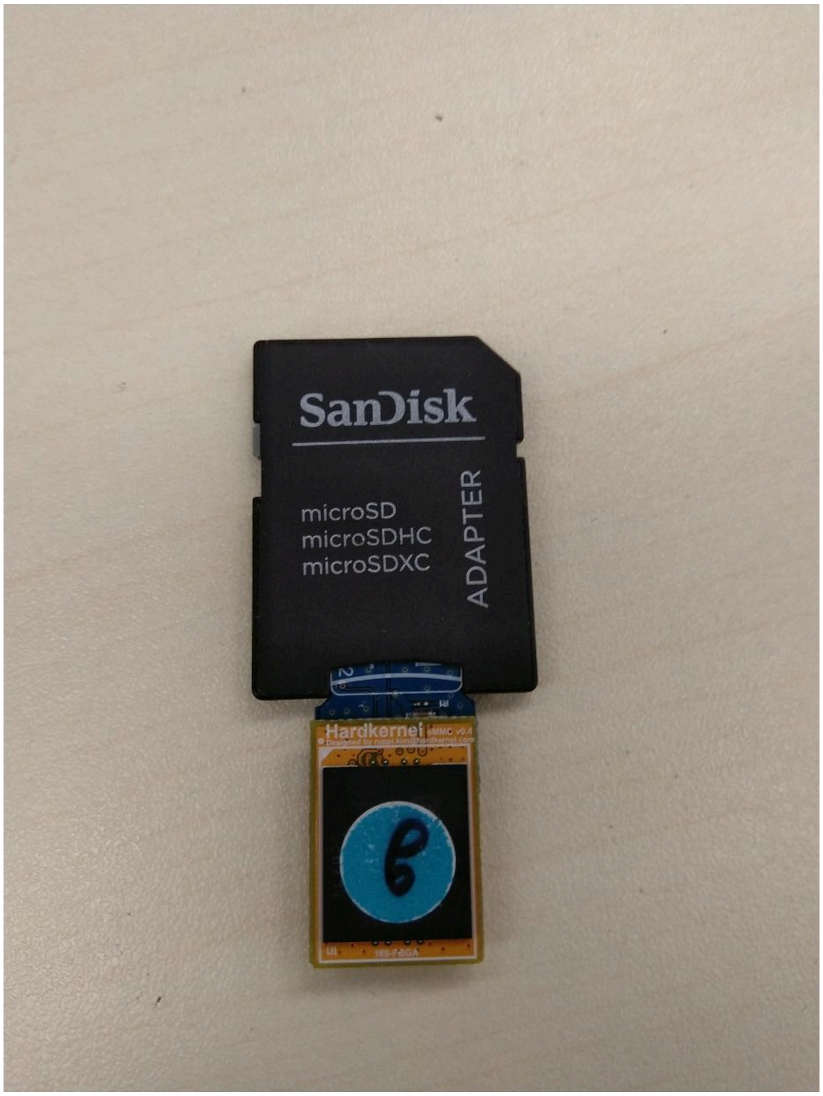
How to test eMMC card on MangoH board

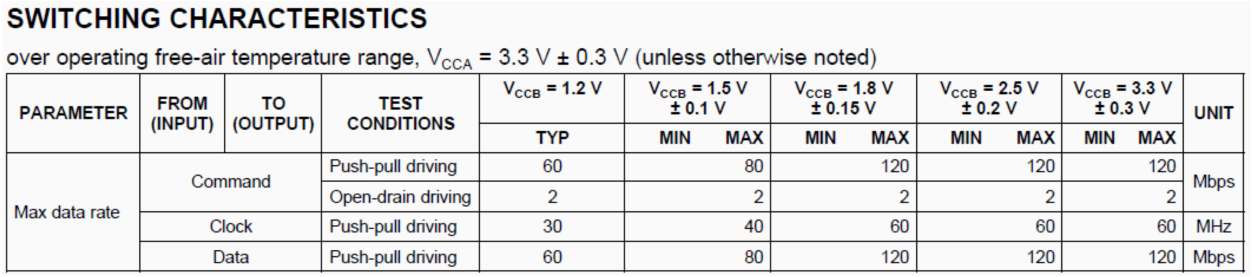
Prerequisite

1. The test below is verified on MangOH Green WP76 module FW R9 with modified yocto image.
2. Make sure the eMMC card with SD card adapter can be read/write in Windows OS or Linux OS before testing on MangOH board.



Procedure

1. For HS200 mode of eMMC standard, the SDIO clock will be running at 200MHz. However the SDIO expander chip used on mangOH Green and Red (TI TXS02612) has a switching capability of 60MHz maximum:



SWITCHING CHARACTERISTICS

over operating free-air temperature range, V_{CCA} = 3.3 V ± 0.3 V (unless otherwise noted)

PARAMETER	FROM (INPUT)	TO (OUTPUT)	TEST CONDITIONS	V_{CCB} = 1.2 V	V_{CCB} = 1.5 V ± 0.1 V		V_{CCB} = 1.8 V ± 0.15 V		V_{CCB} = 2.5 V ± 0.2 V		V_{CCB} = 3.3 V ± 0.3 V		UNIT
				TYP	MIN	MAX	MIN	MAX	MIN	MAX	MIN	MAX	
Max data rate	Command		Push-pull driving	60		80		120		120		120	Mbps
			Open-drain driving	2		2		2		2		2	
	Clock		Push-pull driving	30		40		60		60		60	MHz
	Data		Push-pull driving	60		80		120		120		120	Mbps

As a result, in order to test eMMC card on MangOH board, we need to tune down the SDIO clock to 52MHz when running HS200 mode. Please note that there is no need to do such modification for customer PCB as SDIO communication is directly connected without using SDIO expander.

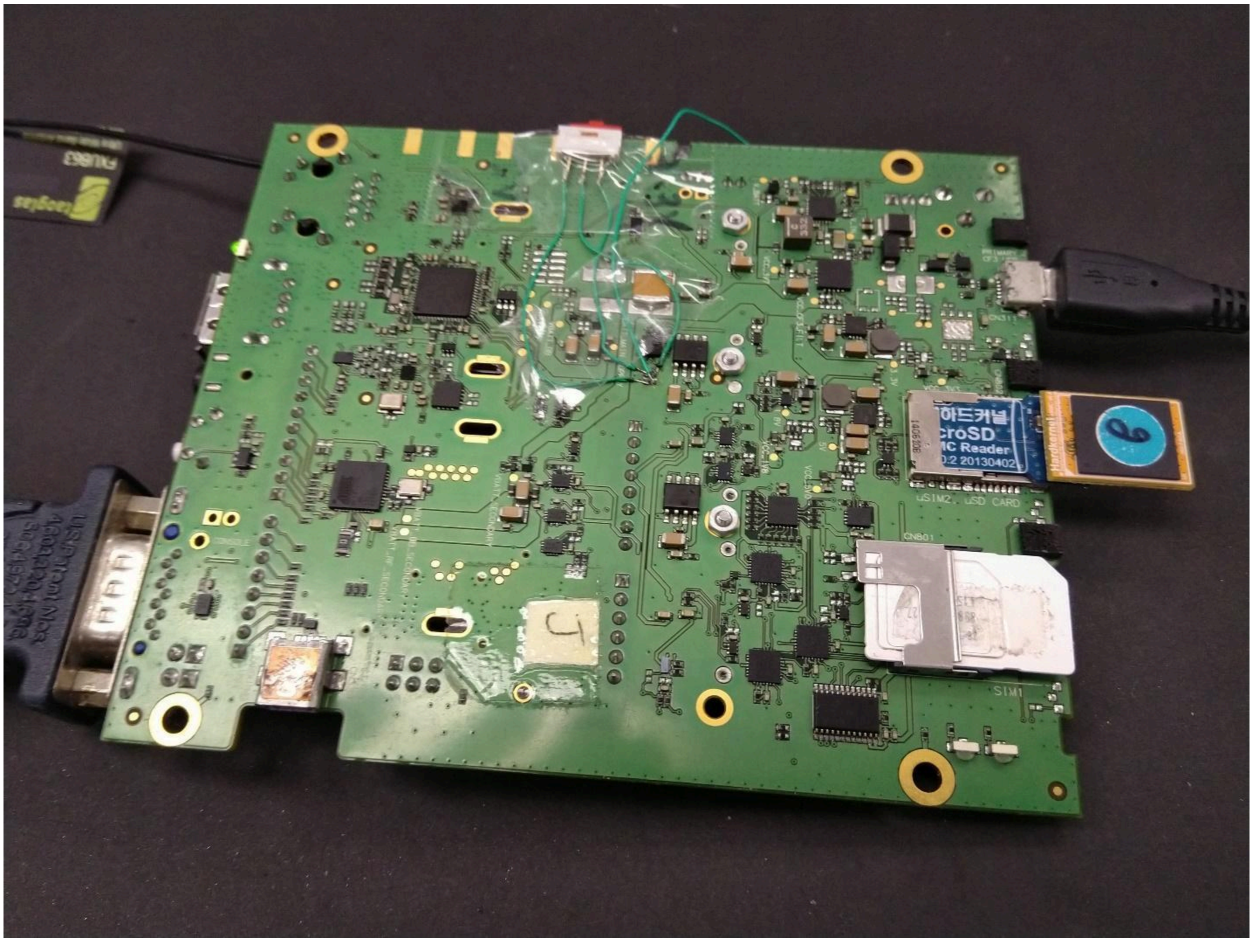
To tune down the SDIO clock to 52MHz when running HS200 mode, modify the yocto source code, /kernel/driver/mmc/core/mmc.c, function mmc_select_card_type() , line around 286, like below:

```
#if 0

        if (caps2 & MMC_CAP2_HS200_1_8V_SDR &&
            card_type & EXT_CSD_CARD_TYPE_HS200_1_8V) {
                hs200_max_dtr = MMC_HS200_MAX_DTR;
                avail_type |= EXT_CSD_CARD_TYPE_HS200_1_8V;
        }
#else

        if (caps2 & MMC_CAP2_HS200_1_8V_SDR &&
            card_type & EXT_CSD_CARD_TYPE_HS200_1_8V) {
                    hs200_max_dtr = MMC_HIGH_52_MAX_DTR;
                    avail_type |= EXT_CSD_CARD_TYPE_HS200_1_8V;
          }
#endif
```

Here I have compiled a yocto image with such modification on top of FW R9.

yocto_wp76xx.4k.cwe

2. Upgrade the modified yocto image to your module.
3. Before power on the module, insert the eMMC card to SD card slot.



4. Now power on the board, after boot into console, you should be able to see the new partition.

```
root@swi-mdm9x28:~# ls /dev/mmc*

/dev/mmcblk0     /dev/mmcblk0p1    /dev/mmcblk0p2    /dev/mmcblk0rpmb
```

5. Also you should be able to see the eMMC detection on kernel message.

```
root@swi-mdm9x28:~# dmesg | grep 'mmc'


[    1.125229] sdhci_msm 7864900.sdhci: No vmmc regulator found

[    1.125246] sdhci_msm 7864900.sdhci: No vqmmc regulator found

[    1.125653] mmc0: SDHCI controller on 7864900.sdhci [7864900.sdhci] using 32-bit ADMA in legacy
mode

[    1.402019] mmc0: jyijyi mmc_select_card_type caps=0x400fa6af caps2=0x383a0021 card_type=0x57

[    1.402042] mmc0: Out-of-interrupt timeout is 50[ms]

[    1.402051] mmc0: BKOPS_EN equals 0x2

[    1.402061] mmc0: eMMC FW version: 0x01

[    1.402070] mmc0: cache barrier support 0 flush policy 0

[    1.411997] mmc0: new HS200 MMC card at address 0001

[    1.416097] mmcblk0: mmc0:0001 8GME4R 7.28 GiB

[    1.416367] mmcblk0rpmb: mmc0:0001 8GME4R partition 3 512 KiB

[    1.417890]  mmcblk0: p1 p2
```

6. You can try to mount the eMMC partition by the following commands:

```
mkdir -p /tmp/eMMC

/bin/mount -t auto -o sync /dev/mmcblk0p1 "/tmp/eMMC"
```

7. To check the SDIO clock speed during read/write operation, you can type the following:

```
root@swi-mdm9x28:~# cat /sys/kernel/debug/mmc0/ios

clock:         50000000 Hz

actual clock:   50000000 Hz

vdd:           18 (3.0 ~ 3.1 V)
```

```
bus mode:       2 (push-pull)
chip select:    0 (don't care)
power mode:     2 (on)
bus width:      2 (4 bits)
timing spec:    9 (mmc HS200)
signal voltage: 0 (1.80 V)
```

8. To unmount the eMMC partition, type

```
umount /tmp/eMMC
```

Reference

https://issues.sierrawireless.com/browse/FORE-47