a module solution provider

i.MX7Dual L4.1.15_2.0.0

WL8 integration Release Note

Date: 2017/01/10

Version History

Date	Version	Remark
2017/01/10	Ver.0.1	The initial version
2017/01/11	Ver.0.2	Add HW preparation
2017/01/13	Ver0.3	Update BT firmware to 18xx_BT_Service_Pack_3.9

1. Preparation

SDIO: mmc1/sdhc2
UART: UART5
GPIO: WLAN_EN: *GPIO4_IO21*
WLAN_IRQ: *GPIO4_IO20*
BT_EN: *GPIO4_IO23*

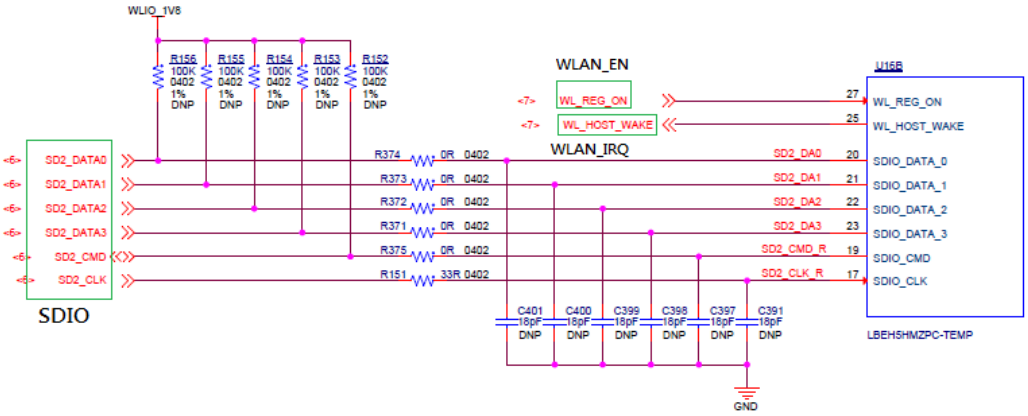
1.1. SW information

Host OS: Ubuntu12.04 64bit
BSP Version: fsl-yocto-L4.1.15_2.0.0-ga
GCC Version: gcc-5.3.0
Kernel Version: Linux4.1.15
WL8 driver Version: R8.6_SP1
Wl18xx firmware version: Rev 8.9.0.1.55
Bluez version: 5.37
Bluetooth Firmware Version : 18xx_BT_Service_Pack_3.9

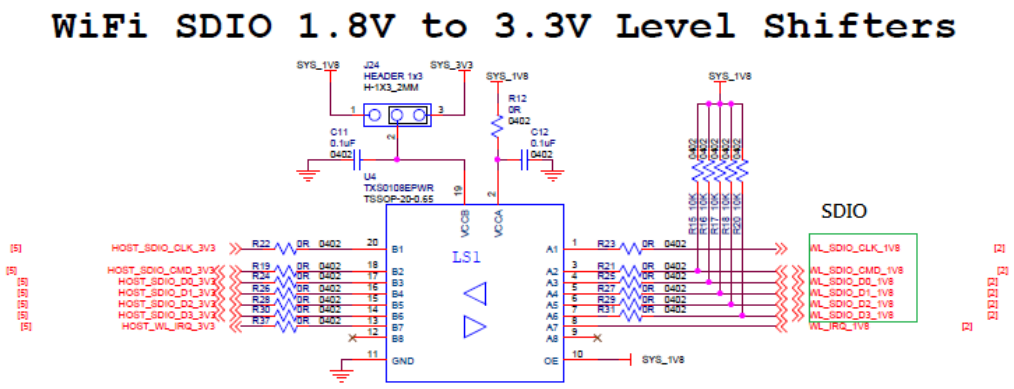
1.2. HW preparation

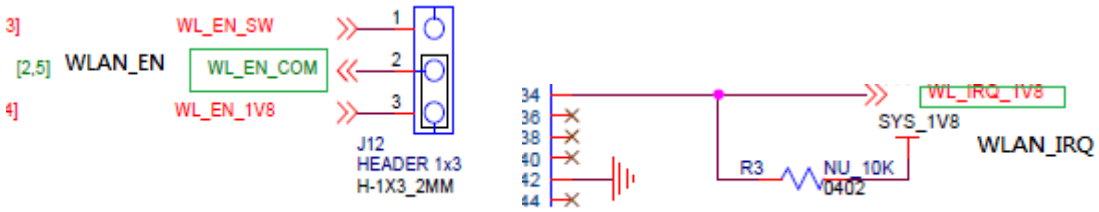
1.2.1.1. wifi

i.MX7:



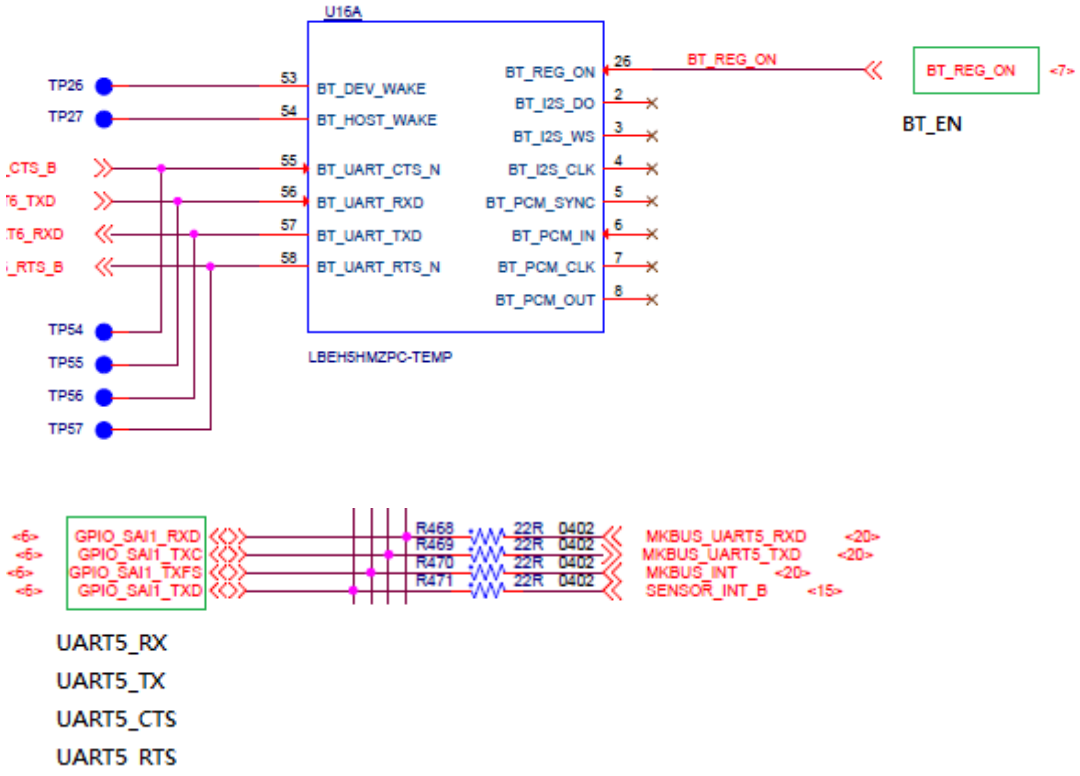
COM82SDMMC:



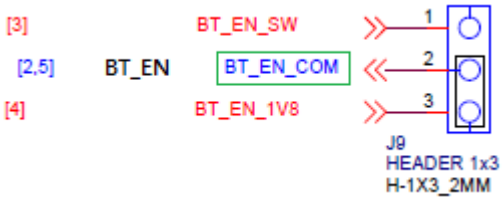


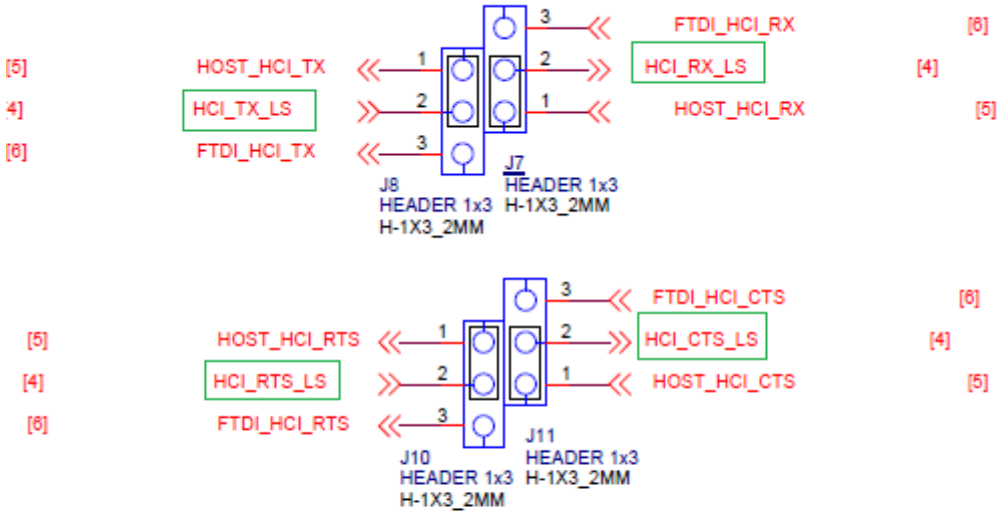
1.2.1.2. BT

i.MX7:



COM82SDMMC:





Host mode	Debug mode (FTDI)	Host mode (Swapped RX,TX and RTS,CTS)
J7,J8,J10,J11 -(short 1-2)	J7,J8,J10,J11 -(short 2-3)	J7-1 to J8-2 / J8-1 to J1-2 J10-1 to J11-2 / J11-1 to J12-2

Note:

Host UART5_TX connected to J8 HCI_TX_LS

Host UART5_RX connected to J7 HCI_RX_LS

Host UART5_CTS connected to J10 HCI_RTS_LS

Host UART5_RTS connected to J11 HCI_CTS_LS

Each jumper would swap the connection to the wl18xx module.

So the final connection is:

i.MX7		wl18xx
UART5_TX	→	HCI_RX
UART5_RX	→	HCI_TX
UART5_CTS	→	HCI_CTS
UART5_RTS	→	HCI_RTS

1.3. Get BSP from NXP

To get the BSP you need to have `repo` installed.

Install the `repo` utility: (only need to do this once):

$: mkdir ~/bin

$: curl http://commondatastorage.googleapis.com/git-repo-downloads/repo > ~/bin/repo

$: chmod a+x ~/bin/repo

$: PATH=${PATH}:~/bin

Download the BSP Yocto Project Environment into your directory:

$: mkdir fsl-arm-yocto-bsp

$: cd fsl-arm-yocto-bsp

$: repo init -u git://git.freescale.com/imx/fsl-arm-yocto-bsp.git -b imx-4.1-krogoth

$: repo sync

1.4. Prepare to build

$ DISTRO=fsl-imx-x11 MACHINE=imx7dsabresd source fsl-setup-release.sh -b build-x11

$ bitbake -c compile -f linux-imx –DDD

$ bitbake -c deploy linux-imx –DDD

$ bitbake core-image-minimal

After build out, make sure you can flash image to SD card or eMMC to boot the board.

2. Prepare Linux Kernel and yocto layer

2.1. Apply gcc for yocto environment setup

1. After you have created a BSP image for your target with a bitbake command, execute the following command to prepare for installation of a SDK
"bitbake <bsp-image-name> -c populate_sdk"

e.g. "bitbake core-image-minimal -c populate_sdk"

2. After the command is finished, deploy the sdk.
   a. Go to ../<BSP>/<build dir>/tmp/deploy/sdk
   b. sudo ./<whatever the name of the .sh script file>
   c. Use the default installation location.

3. Go to /opt/<whatever the name>/
   find the "environmentXYZ.sh"
4. Execute the .sh with the "source" command"
5. At this point the shell session should have all necessary environment variables for the cross-compiler toolchain.

2.2. Rebuild linux kernel and yocto system image

Please apply the patch to BSP.

$ bitbake -c compile -f linux-imx –DDD
$ bitbake -c deploy linux-imx –DDD
$ bitbake core-image-minimal

NOTE: Please copy the latest BT firmware to target rootfs
$ cp BT-firmware/ *.bts ${TARGETROOTFS}/lib/firmware

Then please flash the images and rootfs to the board.

An SD card image provides the full system to boot with U-Boot and kernel. To flash an SD card image, run the following command:
$ sudo dd if=<image name>.sdcard of=/dev/sd<partition> bs=1M && sync

3. Function test

3.1. Wifi

After boot bring up the wifi device initially as a station

```
# ifconfig wlan0 up
```

And then use iw tool to scan for available APs

```
# iw wlan0 scan | grep SSID
```

Now connect to the “unsecured” AP of your choice, here named “AP”

```
# iw wlan0 connect “AP”
```

Wait for the connected messages to come back and then request an IP address from the AP.

```
# udhcpc -i wlan0
```

To bring the Wi-fi up as an Access Point with a dhcp server it is necessary to create a DHCP configuration file that will define the IP addresses to use.

Create a file /etc/udhcpd.conf with following contents

```
# Sample udhcpd configuration file (/etc/udhcpd.conf)
# The start and end of the IP lease block
start                10.4.30.40    #default: 192.168.0.20
end          10.4.30.48    #default: 192.168.0.254
# The interface that udhcpd will use
interface     wlan0                #default: eth0
#Examles
opt  dns     8.8.8.8    8.8.4.4 # public google dns servers
option       subnet         255.255.255.0
opt  router 10.4.30.34
option       lease   864000               # 10 days of
```

Now start hostap using the hostapd.conf already on the target. This will bring up a softAP with SSID “test”

```
#hostapd -B /etc/hostapd.conf
```

Bring up the interface with the IP address defined in udhcpd.conf

```
#ifconfig wlan0 10.4.30.34 netmask 255.255.255.0 up
```

Start DHCP server

```
udhcpd /etc/udhcpd.conf
```

Now a station can connect to AP “test”

3.2. Bluetooth

Start bluetooth daemon

root@imx7dsabresd:~# cd /usr/libexec/bluetooth/
root@imx7dsabresd:/usr/libexec/bluetooth# ./bluetoothd &

Use bluetoothctl to bring up bt and pair

root@imx7dsabresd:~# bluetoothctl
[NEW] Controller EC:11:27:72:72:78 BlueZ 5.37 [default]
[bluetooth]# *power on*
[bluetooth]#
(stc): chnl_id list empty :4
(stk) : st_kim_start(stk) :ldisc_install = 1uim:poll broke due to event 10(PRI:2/ERR:8)

uim:read 1 from install

uim:@ st_uart_config
uim: signal received, opening /dev/ttymxc4
uim:@ set_baud_rate
uim:set_baud_rate() done

uim:Setting speed to 3000000
uim:@ read_command_complete
uim: Command complete started
uim:@ read_hci_event
uim: read_hci_event
uim:Command complete done

uim:Speed changing to 3000000, (stc): st_tty_open 1
uim:@ set_custom_baud_rate
(stk) :line discipline installeduim:Installed N_TI_WL Line displi
ne
uim:begin polling...
(stk) :TIInit_11.8.32.bts(stk) :change remote baud rate command in firmware
(stk) :skipping the wait event of change remote baud(stc): add_channel_to_table: id
4
(stc): add_channel_to_table: id 2
(stc): add_channel_to_table: id 3
Changing power on succeeded
[CHG] Controller EC:11:27:72:72:78 Powered: yes
[bluetooth]# *agent on*
Agent registered
[bluetooth]# *default-agent*
Default agent request successful
[bluetooth]# *pairable on*
Changing pairable on succeeded
[bluetooth]# *discoverable on*
Changing discoverable on succeeded
[CHG] Controller EC:11:27:72:72:78 Discoverable: yes
[bluetooth]# *scan on*
Discovery started
[CHG] Controller EC:11:27:72:72:78 Discovering: yes
[NEW] Device 50:2E:5C:A8:34:F1 HTC Butterfly s
[bluetooth]# *pair 50:2E:5C:A8:34:F1*
Attempting to pair with 50:2E:5C:A8:34:F1
[CHG] Device 50:2E:5C:A8:34:F1 Connected: yes
Request confirmation
[agent] Confirm passkey 259726 (yes/no): yes

[CHG] Device 50:2E:5C:A8:34:F1 Modalias: bluetooth:v000Fp1200d1436
[CHG] Device 50:2E:5C:A8:34:F1 UUIDs: 00001105-0000-1000-8000-00805f9b34fb
[CHG] Device 50:2E:5C:A8:34:F1 UUIDs: 00001106-0000-1000-8000-00805f9b34fb
[CHG] Device 50:2E:5C:A8:34:F1 UUIDs: 0000110a-0000-1000-8000-00805f9b34fb
[CHG] Device 50:2E:5C:A8:34:F1 UUIDs: 0000110c-0000-1000-8000-00805f9b34fb
[CHG] Device 50:2E:5C:A8:34:F1 UUIDs: 0000110e-0000-1000-8000-00805f9b34fb
[CHG] Device 50:2E:5C:A8:34:F1 UUIDs: 00001112-0000-1000-8000-00805f9b34fb
[CHG] Device 50:2E:5C:A8:34:F1 UUIDs: 00001115-0000-1000-8000-00805f9b34fb
[CHG] Device 50:2E:5C:A8:34:F1 UUIDs: 00001116-0000-1000-8000-00805f9b34fb
[CHG] Device 50:2E:5C:A8:34:F1 UUIDs: 0000111f-0000-1000-8000-00805f9b34fb
[CHG] Device 50:2E:5C:A8:34:F1 UUIDs: 0000112f-0000-1000-8000-00805f9b34fb
[CHG] Device 50:2E:5C:A8:34:F1 UUIDs: 00001132-0000-1000-8000-00805f9b34fb
[CHG] Device 50:2E:5C:A8:34:F1 UUIDs: 00001200-0000-1000-8000-00805f9b34fb
[CHG] Device 50:2E:5C:A8:34:F1 UUIDs: 00001800-0000-1000-8000-00805f9b34fb
[CHG] Device 50:2E:5C:A8:34:F1 UUIDs: 00001801-0000-1000-8000-00805f9b34fb
[CHG] Device 50:2E:5C:A8:34:F1 UUIDs: 00006675-7475-7265-6469-616c62756d70
[CHG] Device 50:2E:5C:A8:34:F1 Paired: yes
Pairing successful
[CHG] Device 50:2E:5C:A8:34:F1 Connected: no
[bluetooth]# *trust 50:2E:5C:A8:34:F1*
[CHG] Device 50:2E:5C:A8:34:F1 Trusted: yes
Changing 50:2E:5C:A8:34:F1 trust succeeded
[CHG] Device 50:2E:5C:A8:34:F1 RSSI: -67
[bluetooth]# *quit*
Agent unregistered
[DEL] Controller EC:11:27:72:72:78 BlueZ 5.37 [default]

root@imx7dsabresd:/usr/libexec/bluetooth#

Use opp to transfer file to remote paired device

```
root@imx7dsabresd:/usr/libexec/bluetooth# sdptool browse 50:2E:5C:A8:34:F1
Browsing 50:2E:5C:A8:34:F1 ...
Service RecHandle: 0x10000
Service Class ID List:
  "Generic Attribute" (0x1801)
Protocol Descriptor List:
  "L2CAP" (0x0100)
    PSM: 31
  "ATT" (0x0007)
    uint16: 0x0001
    uint16: 0x0005

Service Name: Headset Gateway
Service RecHandle: 0x10003
Service Class ID List:
  "Headset Audio Gateway" (0x1112)
  "Generic Audio" (0x1203)
Protocol Descriptor List:
  "L2CAP" (0x0100)
  "RFCOMM" (0x0003)
    Channel: 2
Profile Descriptor List:
  "Headset" (0x1108)
    Version: 0x0102

Service Name: Handsfree Gateway
Service RecHandle: 0x10004
Service Class ID List:
  "Handsfree Audio Gateway" (0x111f)
  "Generic Audio" (0x1203)
Protocol Descriptor List:
  "L2CAP" (0x0100)
  "RFCOMM" (0x0003)
    Channel: 3
Profile Descriptor List:
```

"Handsfree" (0x111e)
Version: 0x0106

Service Name: OBEX File Transfer
Service RecHandle: 0x10005
Service Class ID List:
"OBEX File Transfer" (0x1106)
Protocol Descriptor List:
"L2CAP" (0x0100)
"RFCOMM" (0x0003)
Channel: 4
"OBEX" (0x0008)
Profile Descriptor List:
"OBEX File Transfer" (0x1106)
Version: 0x0103

Browsing 50:2E:5C:A8:34:F1 ...
Service Search failed: Invalid argument
Service Name: AV Remote Control Target
Service RecHandle: 0x10006
Service Class ID List:
"AV Remote Target" (0x110c)
Protocol Descriptor List:
"L2CAP" (0x0100)
PSM: 23
"AVCTP" (0x0017)
uint16: 0x0102
Profile Descriptor List:
"AV Remote" (0x110e)
Version: 0x0103

Service Name: Advanced Audio Source
Service RecHandle: 0x10007
Service Class ID List:
"Audio Source" (0x110a)
Protocol Descriptor List:
"L2CAP" (0x0100)

```
      PSM: 25
   "AVDTP" (0x0019)
      uint16: 0x0102
Profile Descriptor List:
   "Advanced Audio" (0x110d)
      Version: 0x0102

Service RecHandle: 0x10008
Service Class ID List:
   "AV Remote" (0x110e)
Protocol Descriptor List:
   "L2CAP" (0x0100)
      PSM: 23
   "AVCTP" (0x0017)
      uint16: 0x0102
Profile Descriptor List:
   "AV Remote" (0x110e)
      Version: 0x0103

Service Name: Android Network Access Point
Service Description: NAP
Service RecHandle: 0x10009
Service Class ID List:
   "Network Access Point" (0x1116)
Protocol Descriptor List:
   "L2CAP" (0x0100)
      PSM: 15
   "BNEP" (0x000f)
      Version: 0x0100
      SEQ8: 0 6
Language Base Attr List:
   code_ISO639: 0x656e
   encoding:       0x6a
   base_offset: 0x100
Profile Descriptor List:
   "Network Access Point" (0x1116)
      Version: 0x0100
```

Service Name: OBEX Phonebook Access Server
Service RecHandle: 0x1000c
Service Class ID List:
"Phonebook Access - PSE" (0x112f)
Protocol Descriptor List:
"L2CAP" (0x0100)
"RFCOMM" (0x0003)
Channel: 19
"OBEX" (0x0008)
Profile Descriptor List:
"Phonebook Access" (0x1130)
Version: 0x0101

Service Name: OBEX Object Push
Service RecHandle: 0x1000d
Service Class ID List:
"OBEX Object Push" (0x1105)
Protocol Descriptor List:
"L2CAP" (0x0100)
"RFCOMM" (0x0003)
Channel: 12
"OBEX" (0x0008)
Profile Descriptor List:
"OBEX Object Push" (0x1105)
Version: 0x0100

root@imx7dsabresd:/usr/libexec/bluetooth# **obexftp -b 50:2E:5C:A8:34:F1 -B 12 -U NONE -p Jorjin.jpg**
Suppressing FBS.
Connecting..\done
Tried to connect for 59ms
Sending "Jorjin.jpg"...\done
Disconnecting..|done