APPLICATION NOTE:
SX-ULPGN-BTZ
Development
Quick Start Guide

Silex Technology America
1751 E. Garry Ave.
Santa Ana, CA 92705
Revision 2.1, Nov 2021

Revision History

Rev. No.	Date	Revision by	Description
1.0	Mar 22, 2019	T.Nakase	Initial draft
1.1	Apr 1, 2019	T.Nakase	Section 3 and Appendix B, EVB_V01 update
1.2	Apr 23, 2019	T.Nakase	Section 6, Jumper setting update/correction
1.3	Aug 23, 2019	T.Nakase	Section 4.2 OpenOCD availability note, Section 5 BDF update, Section 6.2 minor wording change
1.4	Aug 29, 2019	T.Nakase	Section 4.6 Winbond SPI flash note
1.5	Feb 10, 2020	T.Nakase	EVB_V02 update, 3.3V Power Supply jumpers. QCA4020 SDK update.
2.0	Jul 26, 2021	T.Nakase	QCA4020 SDK update. QDN authorized access requirement. Eclipse IDE support discontinued.
2.1	Nov 29, 2021	T.Nakase	Section 6.2 Remove "Gender Changer" requirement.

Table of Contents

1. Scope

The purpose of this document is to provide instructions for setting up firmware development environment of SX-ULPGN-BTZ EVK on Windows 10 with Intel Architecture 64bit CPU. This document assumes the customer has an authorized access to Qualcomm Developer Network (QDN).

2. References

2.1 QDN Document

2.1.1 QCA402x Programming Guide, 80-Y9381-2 Rev.L, Section 3

https://developer.qualcomm.com/hardware/qca4020-qca4024/tools-qca4020

2.2 Software Download Links

2.2.1 GNU Arm Embedded Toolchain: 6-2017-q2-update

https://developer.arm.com/open-source/gnu-toolchain/gnu-rm/downloads

2.2.2 xPack OpenOCD v0.10.0-7

https://xpack.github.io/openocd/releases/

2.2.3 Python: v3.7.6

https://www.python.org/downloads/

2.2.4 QCA4020 SDK: QCA4020.OR.3.4 QCA OEM SDK+CDB r00002.1

https://developer.qualcomm.com/hardware/qca4020-qca4024/tools-qca4020

2.2.5 Zadig v2.5

https://zadig.akeo.ie

3. Equipment

3.1 Hardware

- The EVK board, SX-ULPGN-BTZ EVK (WCBN3516A_EVB V02)
- Host PC
- USB 2.0 Cable (Type A male – Type A male) x2
- Jumper Cap x14
- Jumper Cable (female – female) x3

3.2 Host PC Configuration

- Intel Core i7-1185G7 Processor @ 3.00 GHz
- 16 GB RAM
- 500 GB HDD
- USB 2.0/3.0 x2
- Gigabyte Ethernet Port x1
- Windows 10 Professional
- Username: silex, Account type: Administrator

4. Software Install

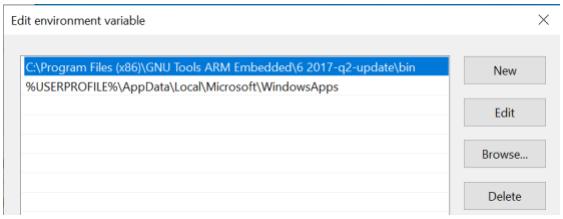
4.1 Toolchain

1. Download **gcc-arm-none-eabi-6-2017-q2-update-win32-sha2.exe** from the download link (Section 2.2.1 above.)

2. Run installer with default settings.

3. Add following path to **Path** user variables[1]:

```
C:\Program Files (x86)\GNU Tools ARM Embedded\6 2017-q2-update\bin
```



Note: QDN Document recommends GCC v6.2 but it is not available on the internet as of writing. The installer package used in this document will install GCC v6.3.1.

4.2 Debugger

1. Download **gnu-mcu-eclipse-openocd-0.10.0-7-20180123-1217-win64.zip** from the download link (Section 2.2.2 above.)

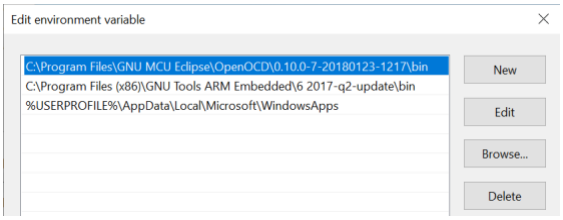
2. Extract it to:

[1] If your account doesn't have Path variable, please click New button and create it as new variable.

```
C:\Program Files
```

3. And add following path to **Path** user variables:

```
C:\Program Files\GNU MCU Eclipse\OpenOCD\0.10.0-7-20180123-1217\bin
```



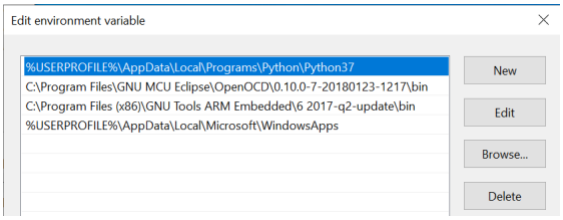
4.3 Python

1. Download **python-3.7.6.amd64.msi** from the download link (Section 2.2.3 above.)

2. Run installer with default settings.

3. Add following path to **Path** user variables:

```
%USERPROFILE%\AppData\Local\Programs\Python\Python37
```



4.4 QCA4020 SDK

1. Download QCA4020.OR.3.4 QCA OEM SDK+CDB from the download link (Section 2.2.4 above.)

2. Extract it to:

```
C:\Users\silex\qca4020-or-3.4_qca_oem_sdk-cdb-r00002.1
```

This document refers this directory as **<SDK_source>** hereafter.

5. Software Configuration

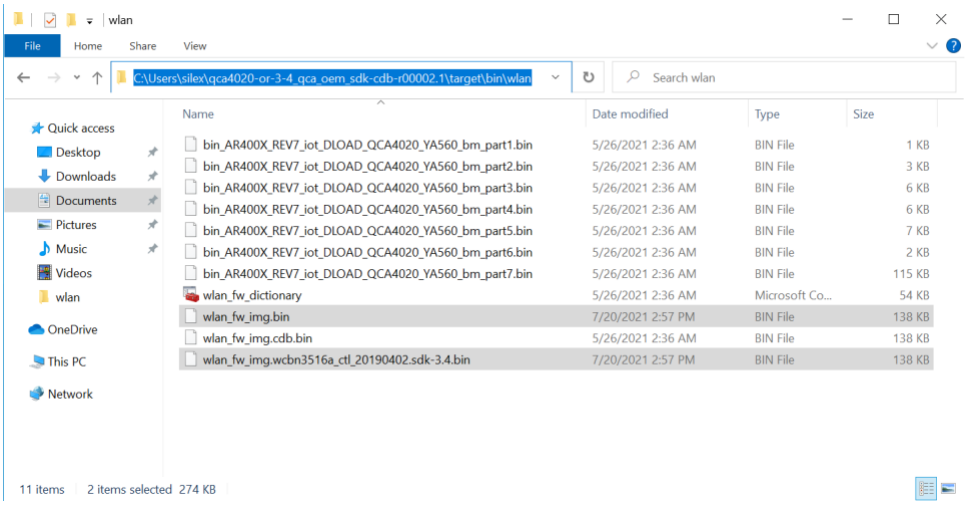
5.1 WiFi Board Data File

1. Copy updated Board Data File (BDF) **wlan_fw_img.wcbm3516a_ctl_20190402.sdk-3.4.bin** to:

```
<SDK_source>\target\bin\wlan
```

2. Backup original BDF **wlan_fw_img.bin** as **wlan_fw_img.cdb.bin**.

3. Overwrite original BDF with updated BDF.



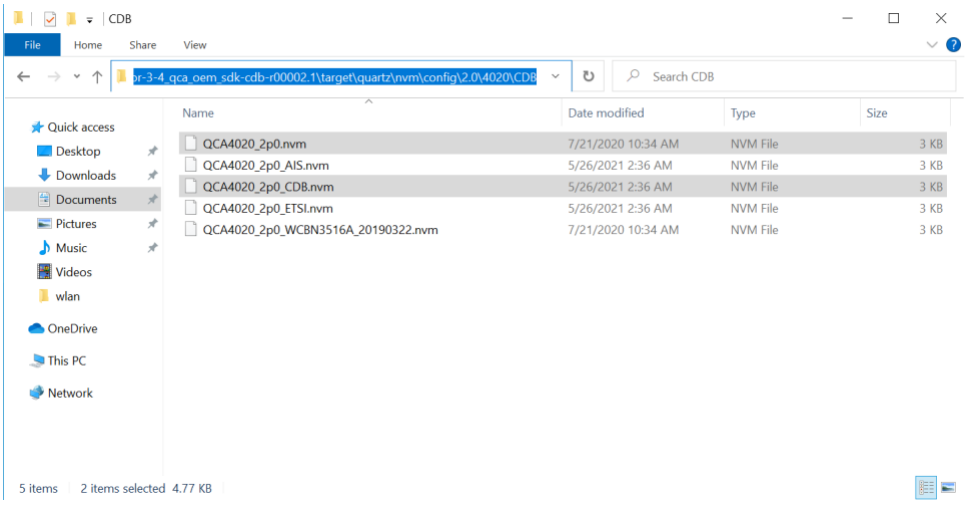
5.2 Bluetooth LE and IEEE 802.15.4 NVM Configuration File

1. Copy updated NVM file **QCA4020_2p0_WCBN3516A_20190322.nvm** to:

   `<SDK_source>\target\quartz\nvm\config\2.0\4020\CDB`

2. Backup original NVM file **QCA4020_2p0.nvm** as **QCA4020_2p0_CDB.nvm**.

3. Overwrite original NVM file with updated NVM file.



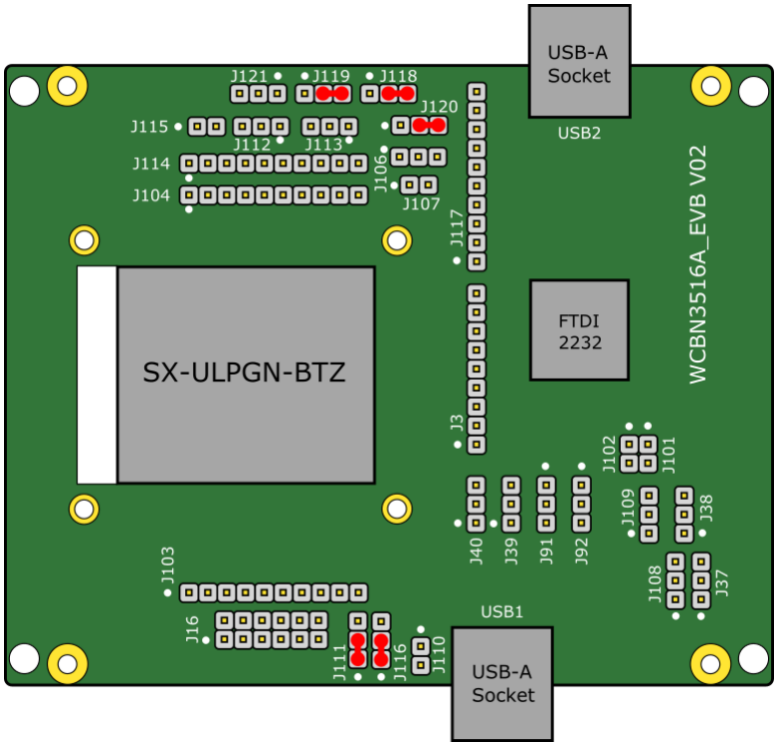
6. Hardware Setup

6.1 Jumper

6.1.1 3.3V Power Supply

Connect following pins to enable **3.3V Power Supply**.

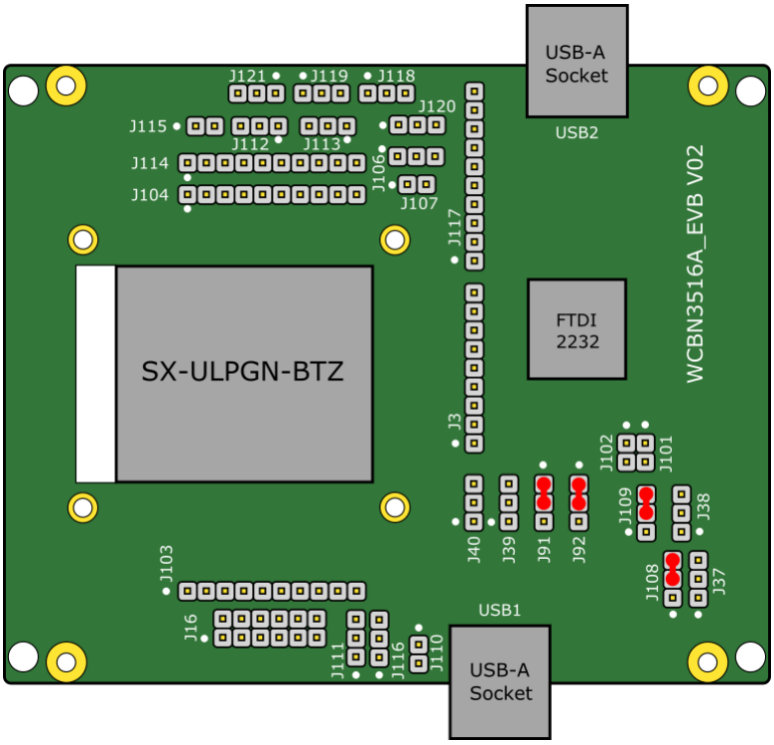
- **J111 Pin 1-2**
- **J116 Pin 1-2**
- **J120 Pin 2-3**
- **J118 Pin 2-3**
- **J119 Pin 2-3**



6.1.2 Debug UART

Connect following pins to enable **Debug UART**.

- **J91 Pin 1-2**
- **J92 Pin 1-2**
- **J108 Pin 2-3**
- **J109 Pin 2-3**

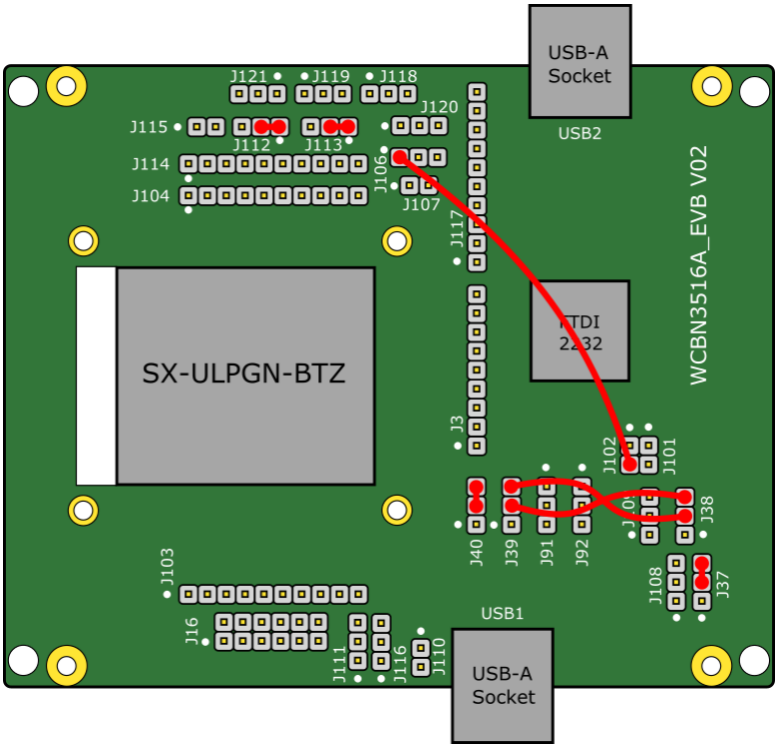


6.1.3 JTAG

Connect following pins to enable **JTAG**.

- **J37 Pin 2-3**
- **J38 Pin 2 - J39 Pin 3 ****
- **J39 Pin 2 - J38 Pin 3 ****
- **J40 Pin 2-3**
- **J102 Pin 2 - J106 Pin 1 ****
- **J112 Pin 1-2**
- **J113 Pin 1-2**

** Needs jumper cable



With this configuration, EVK board bootup in JTAG mode that you can flash and debug custom firmware. To run custom firmware, see Section 9 below.

6.2 Host PC FTDI Driver for JTAG

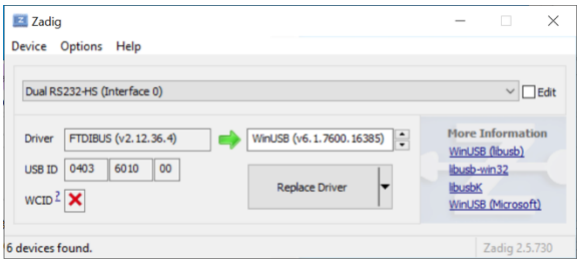
1. Download **zadig-2.5.exe** from the download link (Section 2.2.5 above.)

2. Connect **USB1 and USB2 of EVK board** to **Host PC** with **USB cables**.

3. Start **zadig-2.5.exe**.

4. From top menu, enable **Options > List All Devices** and look for **Dual RS232-HS (Interface 0)**.

5. Select **WinUSB** from the list and click **Replace Driver** button.



7. Build Sample Demo Project: QCLI_demo

7.1 Install Device Configuration File

1. Open **Command Prompt** and go to **<SDK_source>\target\quartz\demo\QCLI_demo\build\gcc**.

2. Run following command:

```
build.bat prepare
```

7.2 Disable Deep Sleep Mode

QCA4020 power down JTAG controller to save power consumption in Deep Sleep mode. To keep JTAG debug session running, disable Deep Sleep mode as follows.

1. Open **target\quartz\demo\QCLI_demo\src\export\DevCfg_master_devcfg_out_cdb.xml** with editor.

2. Set **0 (zero)** to the field shown below:

```
<driver name="Sleep">
  <global_def>
    <var_seq name="devcfgSleepData" type="0x00000003">
        0, 0, 0,
        180, 166, 200,
        end
    </var_seq>
  </global_def>
  <device id="0x02000018">
    <props id="0x1" oem_configurable="false" type="0x00000014"> devcfgSleepData </props>
    <props id="0x2" oem_configurable="false" type="0x00000002"> 0 </props>
    <props id="0x3" oem_configurable="false" type="0x00000002"> 632 </props>
    <props id="0x4" oem_configurable="false" type="0x00000002"> 96 </props>
  </device>
</driver>
```

7.3 Build Demo Project

1. Open **Command Prompt** and go to **<SDK_source>\target\quartz\demo\QCLI_demo\build\gcc** (if you do not open Command Prompt yet.)

2. Run following command:

```
build.bat t 4020 cdb
```

8. Flash Firmware

1. Confirm **3.3V Power Supply** and **JTAG** is enabled (See Section 6 above.)

2. Connect **USB1 and USB2 port of EVK board** to **Host PC** with **USB cables** (if you didn't connect EVK board.)

3. Open **Command Prompt** and go to **<SDK_source>\target\quartz\demo\QCLI_demo\build\gcc** (if you do not open Command Prompt yet.)

4. Run following command:

```
flash_openocd.bat
```

9. Run Firmware

1. Confirm **3.3V Power Supply** and **Debug UART** is enabled (See Section 6 above.)

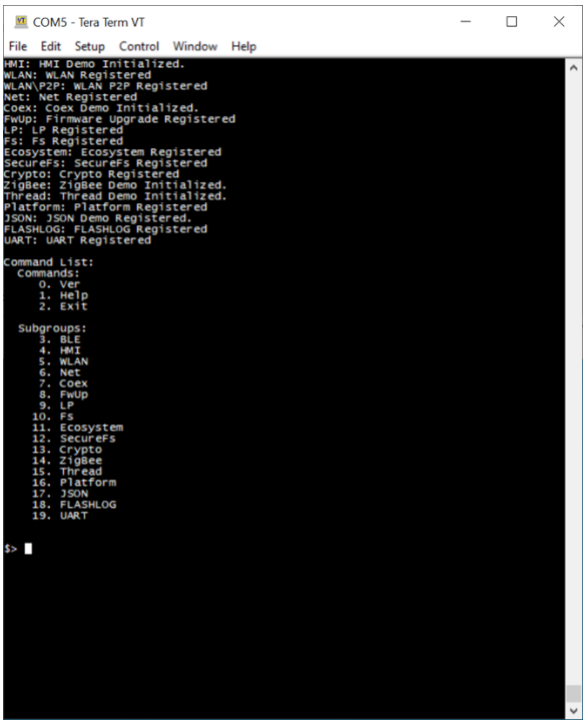
2. Unplug **USB cable** from **USB1 port of EVK board** (if you connected EVK board.)

3. Remove jumper cable from **J102 Pin 2 - J106 Pin 1** to bootup EVK board in **Autoboot mode**.

4. Connect USB cable to **USB1 port of EVK board**.

5. On **Host PC**, open **USB2 port** with console emulator. Baud Rate 115200 bps, Data 8bit, Parity none, Stop Bit 1 bit, and Flow Control none.

6. Connect **USB1 port of EVK board** and you'll see **Command List** on console emulator.



10. Debug Firmware

1. Confirm **3.3V Power Supply** and **JTAG** is enabled (See Section 6 above.)

2. Connect **USB1 and USB2 port of EVK board** to **Host PC** with **USB cables** (if you didn't connect EVK board.)

3. Open **Command Prompt** and go to **<SDK_source>\target\quartz\demo\QCLI_demo\build\gcc** (if you do not open Command Prompt yet.)

4. Run following command:

```
debug_openocd.bat m4
```

The batch file start GDB server and open GDB CLI session.

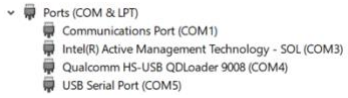
Note: Following preset break points will hit while reaching to app_init().

- Breakpoint 1, _main()
- Breakpoint 2, sbl1_main_ctl()

Appendix A: Emergency Loader (EDL)

Emergency Loader (EDL) mode can program firmware with **USB1 port**. It needs Qualcomm USB driver which available only for Windows.

1. Unplug **USB cables** (if you connected EVK board.)
2. Confirm **3.3V Power Supply** is enabled.
3. Remove jumper connecting **J102 Pin 2 - J106 Pin 1** (if you enabled JTAG.)
3. Connect **J106 Pin 1-2** for **EDL (Emergency Loader) mode**.
4. Connect **USB1 of EVK board** with **USB cable** to **Host PC**.
5. Find **COM port number** with **Device Manager**. Look for **Qualcomm HS-USB QDLoader 9008** under **Ports (COM & LPT)** category, COM4 in the screenshot below.

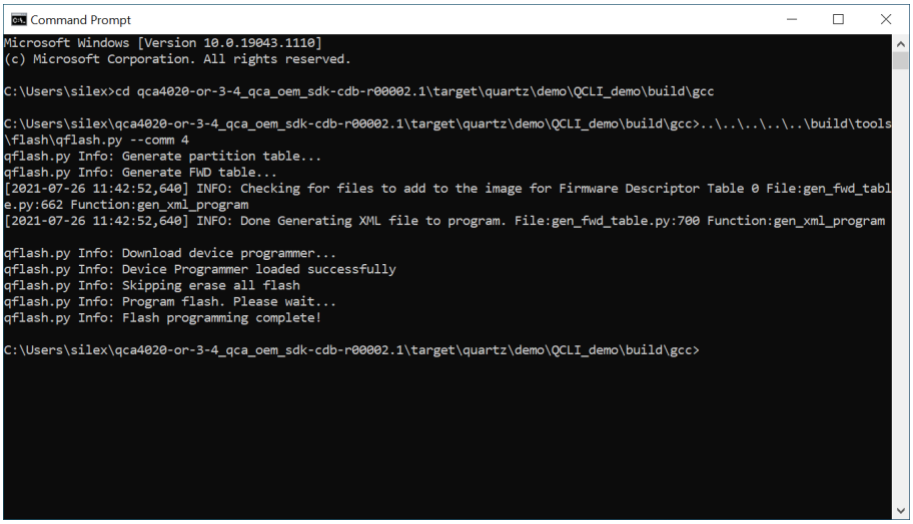


6. Open **Command Prompt** and go to **<SDK_source>\target\quartz\demo\QCLI_demo\build\gcc**.
7. Run following command[2]:

   ```
   ..\..\..\..\..\build\tools\flash\qflash.py --comm 4
   ```

   Specify **COM port number** with --comm option.

8. Wait until you see **Flash programming complete!** message.



9. Remove **J106 Pin 1-2** to restore **Autoboot mode**.

[2] If fh_loader.exe throw the application error (code 0xc000007b), install Visual C++ Redistributable 2010 to Host PC.

Appendix B: What's different from Qualcomm CDB20/M20?

Jumper Header

Category	Description	CDB20		SX-ULPGN-BTZ EVK	
		Ref	Default Position	Ref	Default Position
Power	J6 USB Power Source (Option 1)	J57	Connect pins 2 and 3 for J6 Power	n/a	n/a
	J85 USB Power Source (Option 2)	J57 J36	Connect pins 1 and 2 for J85 Power Connect pins 1 and 2 for J85 Power	n/a	n/a
	LED_PWR	J20	Connect pins 1 and 2 for power up LED	J110	Connect pis 1 and 2 for power up LED
	Vbatt Selection	J22	Connect pins 1 and 2 for fixed +3.3V	n/a	n/a
	Reset Switch	S7	OFF	n/a	n/a
	Power Supply Voltage	n/a	n/a	J111	Connect pins 1 and 2 for 3.3V Power Supply
				J116	Connect pins 1 and 2 for 3.3V Power Supply
				J118	Connect pins 2 and 3 for 3.3V Power Supply
				J119	Connect pins 2 and 3 for 3.3V Power Supply
				J120	Connect pins 2 and 3 for 3.3V Power Supply
EDL	Download Mode	J34	For autoboot mode, remove jumper For EDL mode, connect pins 1 and 2	J106	For autoboot mode, remove jumper For EDL mode, connect pins 1 and 2
LED	LED source	J16	Connect pins 1 and 2 for the WLAN white LED	J16	Connect pins 1 and 2 for the WLAN white LED
			Connect pins 3 and 4 for the 802.15.4 red LED		Connect pins 3 and 4 for the 802.15.4 red LED
			Connect pins 5 and 6 for the Bluetooth blue LED		Connect pins 5 and 6 for the Bluetooth blue LED
			Connect pins 7 and 8 for the red Tri LED		Connect pins 7 and 8 for the red Tri LED
			Connect pins 9 and 10 for the blue Tri LED		Connect pins 9 and 10 for the blue Tri LED
			Connect pins 10 and 12 for the green Tri LED		Connect pins 10 and 12 for the green Tri LED
Sensor	Humidity & Temperature	J17	Connect pins 1 and 2 for Humidity & Temperature	J115	Connect pins 1 and 2 for Humidity & Temperature
Serial	Debug UART (GPIO 9:8)	J18	Connect pins 2 and 3 for Debug UART	J108	Connect pins 2 and 3 for Debug UART
		J19	Connect pins 2 and 3 for Debug UART	J109	Connect pins 2 and 3 for Debug UART
		J91	Connect pins 1 and 2 for Debug UART	J91	Connect pins 1 and 2 for Debug UART
		J92	Connect pins 1 and 2 for Debug UART	J92	Connect pins 1 and 2 for Debug UART
JTAG	4-pin JTAG (GPIO 27:24)	J30	Connect pins 1 and 2 for JTAG	J113	Connect pins 1 and 2 for JTAG
		J31	For autoboot mode, remove jumper For JTAG debug/flashing, connect pins 1 and 2	J102 J106	For autoboot mode, remove jumper For JTAG debug/flashing, connect J102 pin 2 and J106 pin 1
		J32	Connect pins 1 and 2 for JTAG	J112	Connect pins 1 and 2 for JTAG
		J37	Connect pins 2 and 3 for JTAG	J37	Connect pins 2 and 3 for JTAG
		J38 J39	Connect J38 pins 2 and J39 pins 3 for JTAG Connect J38 pins 3 and J38 pins 2 for JTAG	J38 J39	Connect J38 pins 2 and J39 pins 3 for JTAG Connect J38 pins 3 and J38 pins 2 for JTAG

		J40	Connect J40 pins 2 and 3 for JTAG	J40	Connect J40 pins 2 and 3 for JTAG

Debug Header

Debug Header arrangement and assignment has changed. Refer to the schematics of respective evaluation board.

Arduino Header

Qualcomm CDB20 has Arduino compatible header.

External PA

Qualcomm M20 QCA4020 module has external PA to the RF path of Bluetooth LE, ZigBee/Thread/802.15.4. Measured gain is 10-15 dBm over SX-ULPGN-BTZ.

Onboard Sensor

In addition to LED, Humidity and Temperature sensor onboard, Qualcomm CDB20 has following extra set of sensors.

- Pressure
- Accelerometer and Gyro
- Compass and Magnetometer
- Ambient Light
- PIR (with amplifier)