Dear \_\_.

We are facing problems running the SDK on our board.

We selected a simple example to start with - CCG3PA Car Charger (CLA).

We used PSoC  creator 4.2.

The set up was:

* CY4532 EVK Board
* Main board jumpers position:
  + J2 short 1-2,
  + J3: short 1-2
  + J4 short 1-2
  + J6 short 2-3
* Power board J14 short
* Miniprog3 connected
* For load, we connected CY4533 to the CY4532 USB-C connector with USB-C cable.
* Enabled SWD interface in system tab

Step 1: Customized CCG3PA Car Charger (CLA) example PSoC creator 4.2

1. Disabled CCG\_TYPE\_A\_PORT\_ENABLE  in stack\_params.h
2. Disabled Page type A
3. Enabled SWD interface in system tab

Running Debug mode: added three breakpoints in the app event handler function:

 APP\_EVT\_TYPEC\_ATTACH

APP\_EVT\_DISCONNECT

 APP\_EVT\_PD\_CONTRACT\_NEGOTIATION\_COMPLETE

When checked above cases: Pluging and un-plugging USB type c load, the debugger responded correctly, and stopped at the above break points.

Note1 : After programming and running (in debug mode) the CCG3PA Car Charger (CLA) example, we failed to read the device configuration via ez pd configuration utility.

According to an advice I received in one of the forms, I had to program, via PSoC programmer, file CYPD3171-24LQXQ\_cla\_3\_4\_0\_2274\_0\_0\_0\_pa.hex from C:\Program Files (x86)\Cypress\EZ-PD CCGx Power SDK\CCGx\Firmware\binaries\CYPD3171-24LQXQ\_cla. It solved the protocol issue, but destroyed the content I wanted to read.

Q1: How can I read the original content?

Prepared original\_cla.c which represented the configuration table. When I read the file, received "type unknown" in the field \_\_\_\_, instead of UFP-AMA.

Q2: What is the meaning of UFP-AMA, and why did I receive “type unknown”?

Step 2: Add peripherals

* Add SW\_Tx\_UART pin P1.2 ( TP13 )
* Connect LED on pin P2.3 toggled 1 sec.

#define APP\_FW\_LED\_ENABLE (1u)

#define FW\_LED\_GPIO\_PORT\_PIN (GPIO\_PORT\_2\_PIN\_3)

* Added debug print via SW\_Tx\_UART in app.c

APP\_EVT\_TYPEC\_ATTACH

APP\_EVT\_DISCONNECT

APP\_EVT\_PD\_CONTRACT\_NEGOTIATION\_COMPLETE

* When I ran the Firmware, shortened pins 1-2 of J6, the system worked as expected: LED toggled, UART printed as expected when plugs was inserted and unplugged.
* When I repeated the above in debug mode, the LED and UART did not work, and no response after connecting or disconnecting USB-C load

Q3: Why the Firmware did not run in Debug Mode?

Step 3: Running the example on AMS Circuit that is based on CYPD3175

1. Device selector -> CYPD3175

Loaded the firmware with the breakpoints and tried to run.

The firmware did not stop in any breakpoint when I plugged and unplugged the USB-C load.

Loaded the firmware without the additions, only with the break points, and ran in debug mode.

The device initialized and entered Deep Sleep state. Then any plugging or unplugging of the USB-C load, did not “wake” the system (the firmware did not reach any break point).

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Function Name: CySysPmDeepSleep

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*//\*\*

\*

\* Puts the part into the Deep Sleep state. If the firmware attempts to enter

\* this mode before the system is ready (that is, when

\* PWR\_CONTROL.LPM\_READY = 0), then the device will go into the Sleep mode

\* instead and automatically enter the originally intended mode when the

\* holdoff expires.

\*

\* The wakeup occurs when an interrupt is received from a DeepSleep or

\* Hibernate peripheral. For more details, see a corresponding

\* peripheral's datasheet.

\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Q4: What should be changed to enable “wake up” from the deep sleep when plugging and unplugging the USB-C load?

Q5: What is “corresponding peripheral's datasheet” mentioned above, and how can we get it?

Q6: What is required to run CCG3PA Car Charger (CLA) example on CYPD3175? We changed the "device selector " in PSoC 4.2, from CYPD3171 to CYPD3175, but it seems as additional changes are required.

In AMS application, we disable the following functions, because AMS circuit is controlling the DC/DC via I2C. We assume that siabling this functions should not have any effect when we run the firmware till receiving the load requested power level.

|  |  |
| --- | --- |
| pd\_remove\_internal\_fb\_res\_div() | AMS design need to be disable |
| pd\_hal\_set\_vbus\_csa\_rsense(pd\_get\_ptr\_pwr\_tbl(0)->cur\_sense\_res) | AMS design need to be disable |
| pd\_hal\_disable\_vreg (TYPEC\_PORT\_0\_IDX); | AMS design need to be disable |

Q7: Can we disable these functions and run till getting plug and unplug messages?