Application Report
SLVAE18–July 2018

Using I2C Master in TPS65987D and TPS65988 PD Controllers

Rahul Sharma

ABSTRACT

The TPS65987D and TPS65988 are stand-alone USB Type-C and Power Delivery (PD) controller providing cable plug and orientation detection for two USB Type-C connectors. Upon cable detection, these PD controllers communicates on the CC wire using the USB PD protocol. When cable detection and USB PD negotiation are complete, the TPS65987D and TPS65988 enables the appropriate power path and configures alternate mode settings for external multiplexers.

This device also includes three I2C ports out of which one I2C port can work as both master and slave, one I2C port can work as a I2C master only and one can work as a I2C slave only. The I2C masters allow the PD controller to control various kinds of slaves directly instead of using an external controller to do the same. This helps to design robust systems with least external dependencies while reducing the system latencies.

This application report will also demonstrate the usage of TI's configuration utility to configure the I2C masters of TPS65987D and TPS65988 PD Controller to control two different kind of devices.

Contents

1 Introduction 2
2 Advantages of having an I2C master in a PD controller 2
3 How to configure I2C master in TI's Configuration Utility 2
4 Configuring I2C Master for BQ25703A Battery Management Controller (BMC) 7
5 Configuring I2C Master for TUSB546 Alternate Mode Linear Redriver Crosspoint Switch 13
6 Summary 15
7 References 15
Appendix A 16

List of Figures

1 Global System Configuration 3
2 I2C Master Configuration 4
3 Show App Configuration Data Table 5
4 App Configuration Data Table 6
5 App Configuration Binary Data Indices 7
6 Record Index 1 for BQ25703A 8
7 Record Index 2 for BQ25703A 9
8 Record Index 3 for BQ25703A 9
9 Configuration for Hard Reset and Detach events 10
10 Record Index for 5 V Sink PDO 11
11 App Configuration Binary Data Indices for BQ25703A 12
12 Record Index for 5 V Source PDO 13
13 Power Up Data for TUSB546 14
14 Configuration Data for TUSB546 14
15 App Config Binary Data Indices for TUSB546 15

List of Tables

1 I2C Port Features 2

Trademarks

1 Introduction

This application note is aimed to demonstrate the usage of TPS65987D and TPS65988's I2C Master's configuration, it's usage and the how it can be used to solve system complexity.

TPS65987D and TPS65988 each contain three I2C ports which can be used to control an I2C slave, like an alternate mode mux, battery charging controller and others. At the same time, I2C ports can be used as an I2C slave to provide Host Interface to control the various features of the chip. All of these I2C ports can support Fast (400 KHz) and Standard (100 KHz) I2C modes with burst writing. Check the Table 1 to understand the features offered by each if the I2C port.

Table 1. I2C Port Features

Port	Master	Slave	Comments
I2C1	Yes	Yes (Default)	Should be connected to the embedded controller of the PC
I2C2	No	Yes	Should be connected to the Thunderbolt controller of the PC
I2C3	Yes	No	Should be used to control the external I2C slave devices like Mux, Battery Chargers and others.

2 Advantages of having an I2C master in a PD controller

In a traditional system a PD controller is generally an I2C slave and therefore needs an external embedded controller (EC) to control all the external I2C slave devices like battery management controllers, I2C based alternate mode multiplexers and so forth. In such systems, the EC is interrupted when there is a PD event, the EC then goes and queries the PD controller upon receiving the interruption. The EC then processes the received data from the PD controller and then finally controls the required device. As it can be seen, that this is a long process involving multiple I2C read and write transactions, it therefore takes long time and involves high latencies. In certain situations the latencies increases so much that the system timings are not met and the complete PD negotiation has to be retried.

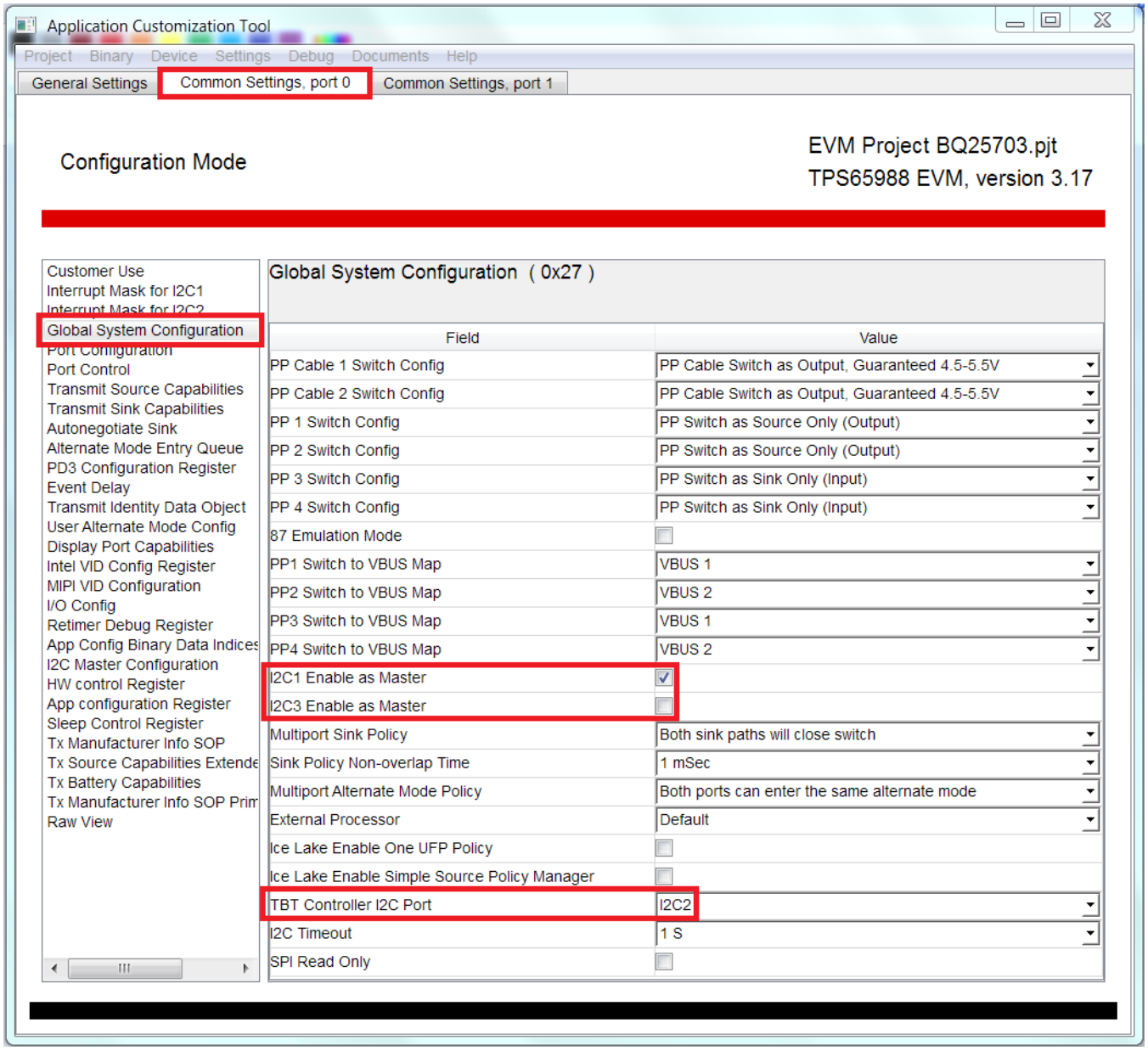
In order to solve this issue the TI's TPS65987D and TPS65988 series of devices comes with two I2C Masters which can be used to directly control external I2C slaves. Controlling these slaves devices directly makes the solution simpler, more efficient, reduces the latencies and makes the system more robust.

3 How to configure I2C master in TI's Configuration Utility

There are multiple steps involved to configure the I2C master which are listed below:

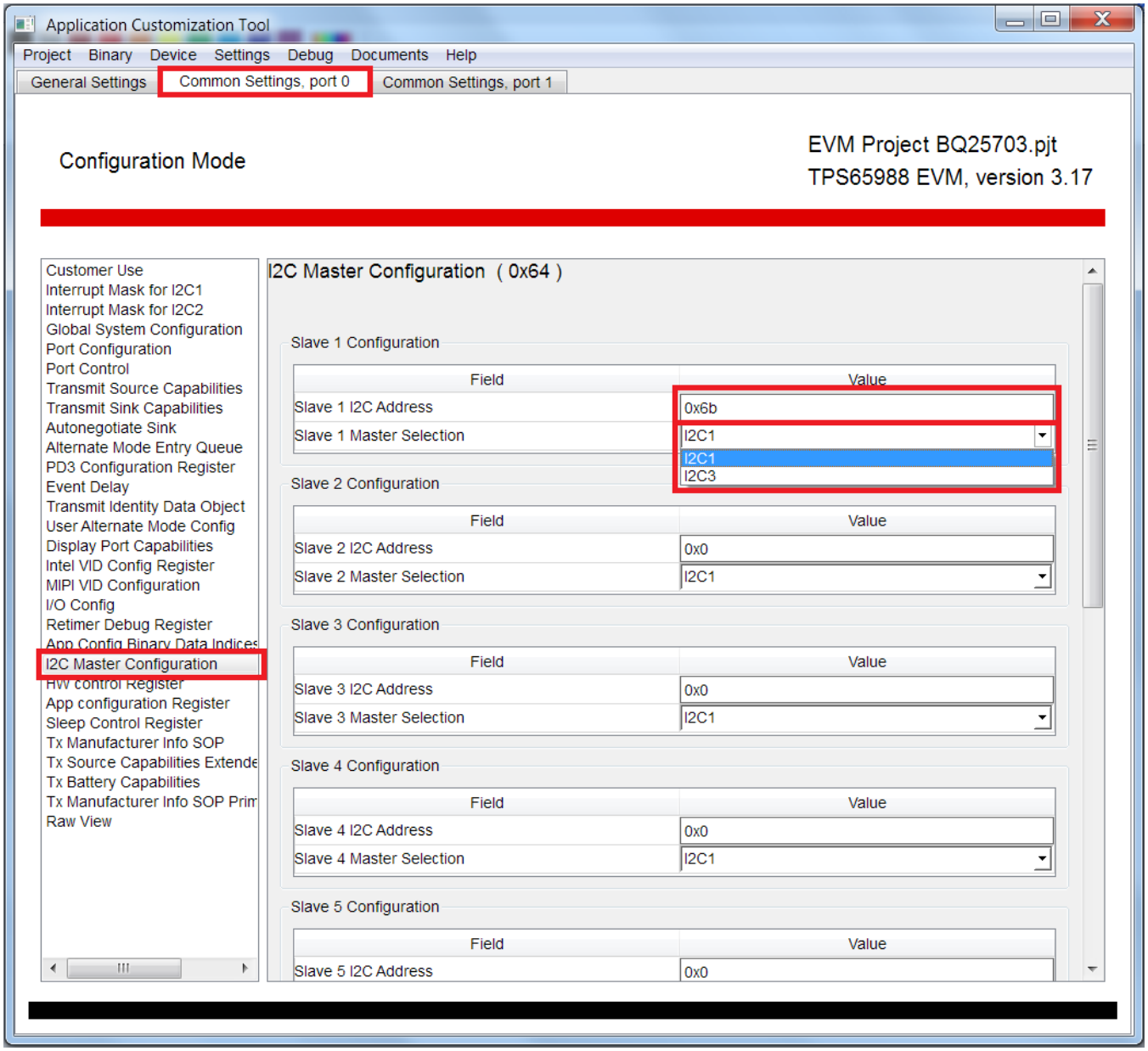
1. Create a project or open a existing project and go to "Global System Configuration" tab.
2. Check the check boxes against the I2C port which you want to use as master. Please note that I2C1 port can't be used for TBT controller if it's used as an I2C master.

Figure 1. Global System Configuration



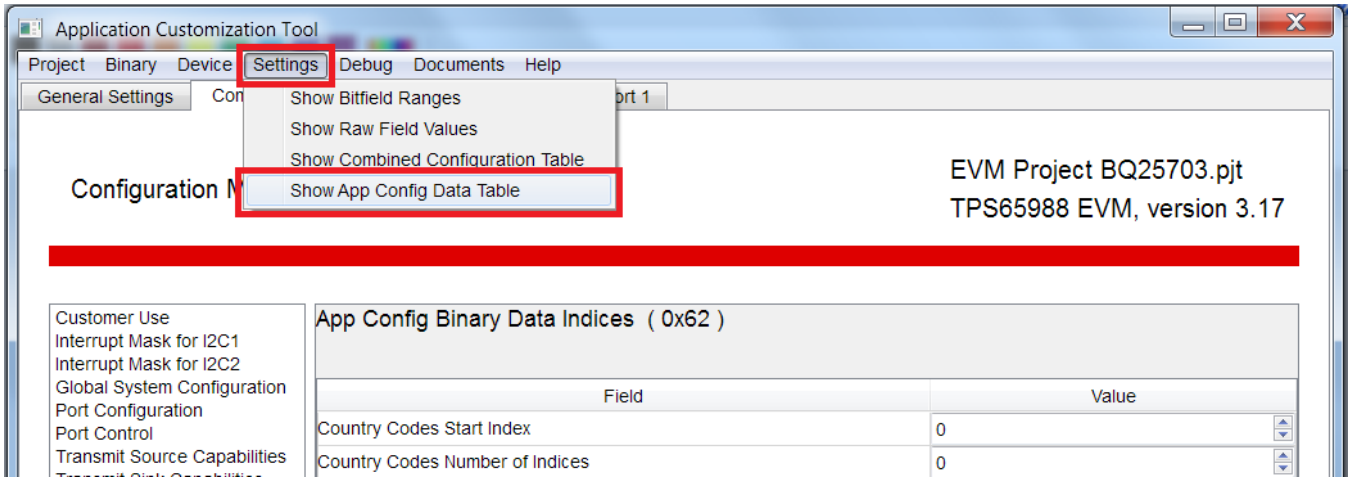
3. Now go to "I2C Master Configuration" tab and set the I2C address of each of the connected I2C slave and the I2C port at which they are connected. Since there are two I2C master ports in TPS65987D and TPS65988 devices, we can choose between the available option by using the drop down menu for each of the slave. Configure this tab for all the slaves you have and leave reamining slave configurations at the default (Slave x I2C Address = 0x0).

Figure 2. I2C Master Configuration



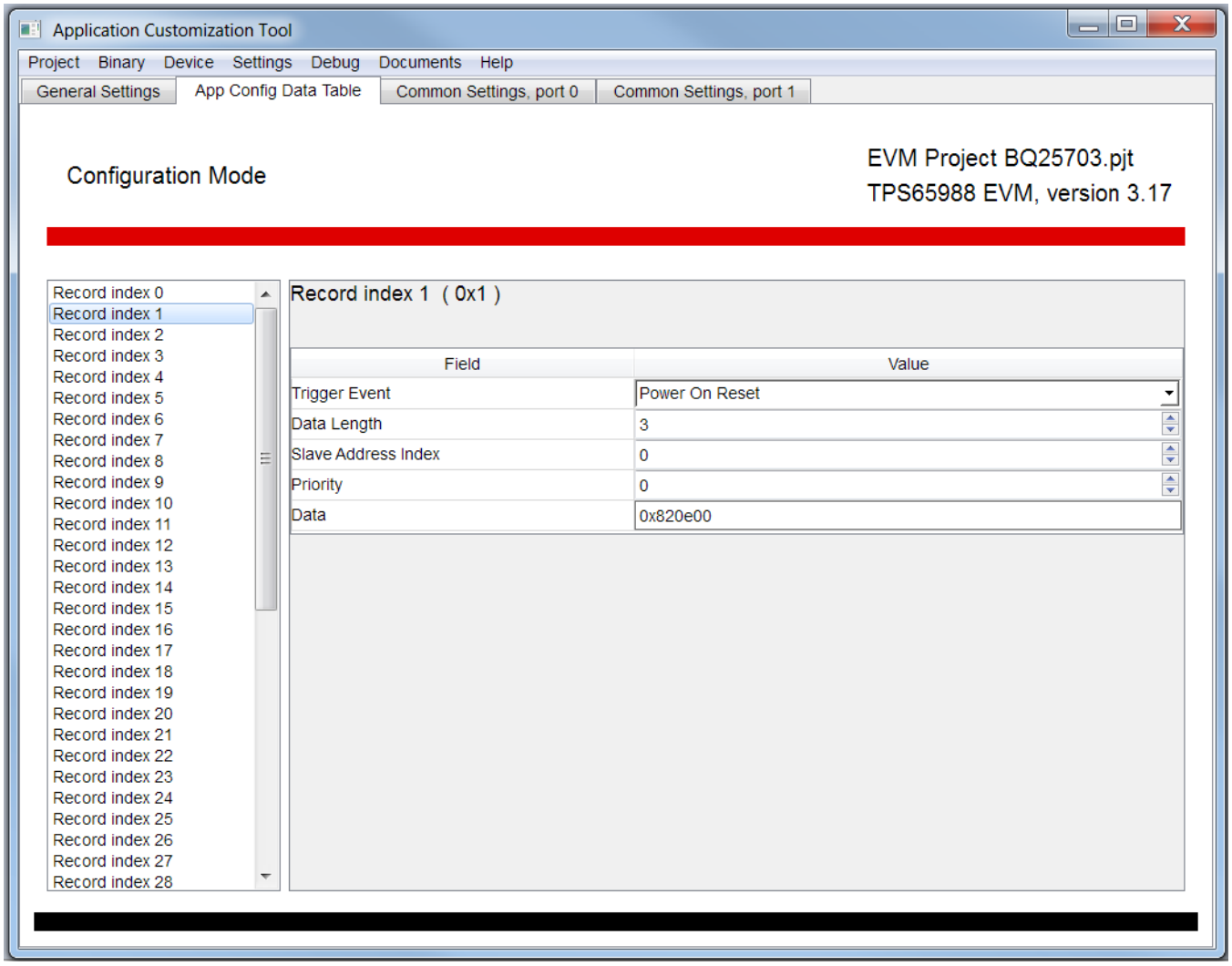
4. Now go to "Settings Menu" and select "Show App Config Data Table" to open the "App Config Data Table". This table contains all the unique hex values which will be send when triggered by an event. Depending on the event the corresponding binary data from the mapped record gets transmitted over I2C when that particular even is triggered.

Figure 3. Show App Configuration Data Table



5. The "App Config Data Table" can take up to fifty records of 10 Bytes of data each. In this table the Record Index 0 is reserved for Country Code and the remaining 49 records can be used by the user for his application. Each of these record index have a trigger event associated with it and when this event happens it sends the data filled in the Data field to the associated slaves. There are five fields in per record (except the Record index 0) which are as follows:
   a. Trigger Event: This field tells when the data stored in this index will be sent out to the I2C slave. Please see the Appendix A for the list of events.
   b. Data Length: This field defines the length of data to be sent. It should match with the number of bytes of data in the Data field
   c. Slave Address Index: This field allows the user to attache this record to a particular I2C slave. In the given example below the Record Index 0 is attached with the first I2C slave.
   d. Priority: Depending on the system requirement, there can be multiple bytes of data to be sent to different I2C Slaves for a given PD event. In such a scenario priority can be assigned to the Record Indexes properly defining which Record Index will be processes earlier in case of conflict. A lower number in this field indicates higher priority.
   e. Data: Values filled here goes out to the I2C slave. This value is always entered in hex and the rightmost bytes goes out first.

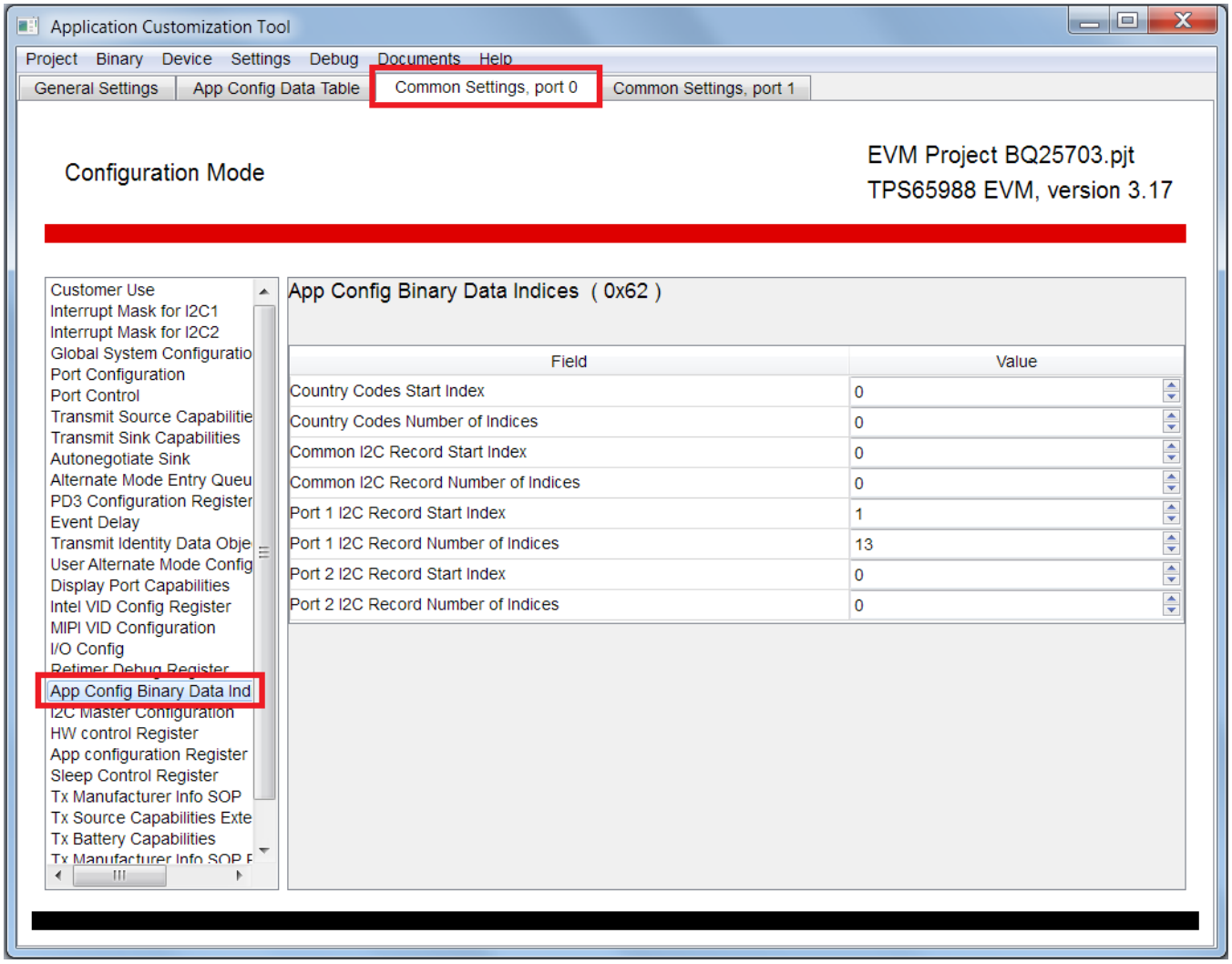
Figure 4. App Configuration Data Table



6. Now go the Common Settings, port 0 tab and fill the "App Config Binary Data indices". There are multiple number which needs to be filled depending on the system requirements and the number of Record Indexes used in the previous steps.

Fill the Common I2C Records for I2C events and slaves which are common to both the ports and for port specific events fill Port x I2C records. Record Start index tell from which index does the data starts and Number of Indexes tell how many records are there for the I2C in general or for each of the ports. In our case the record started from Record Index 1 and we used total of nine Record Indexes. The slave is connected to Port 1 and therefore we filled the data as below image.

Figure 5. App Configuration Binary Data Indices



4 Configuring I2C Master for BQ25703A Battery Management Controller (BMC)

BQ25703A is a synchronous NVDC battery buck-boost charge controller for multi-chemistry battery charging applications. This controller is specifically designed for USB-PD systems covering the complete 20V/5A support. Please refer BQ25703A datasheet for more product details.

This BMC has 60 registers of 8 bits each, out of these 60 registers only 9 registers which needs to be updated. Details of these registers are as follows:

Register Address	Comment
0x00, 0x01	Initial configuration of BQ25703A
0x01, 0x02	Charging Current
0x06, 0x07	OTG Voltage
0x09, 0x09	OTG Current
0x35	Enable OTG Mode

The first two registers of BQ25703A initializes all the internal parameter and needs to be written only once during power up sequence. The register 0x00 needs to have a value of 0x0E and 0x01 should have a value of 0x82. Please see the register map of in the datasheet of BQ25703A for more details.

The registers 0x02 and 0x03 sets the battery charging current for BQ25703A. It can be treated as a single 16 bit register whose every each count represents 1 mA of charging current. But please note that the first 5 bits of the register are reserved and anything written there will have no effect. Therefore the current can be varied in the steps of 64 mA and can go to a maximum of 8.128 A. Whenever the BQ25703A is supposed to charge the battery these registers needs to be programmed to set the battery charging current. These registers might need to be programmed even when there's a power role swap from source to sink and during the change in PDOs.

Registers 0x06, 0x07, 0x08, 0x09 are used to set OTG Voltage and Current. These registers needs to be programmed whenever BQ25703A needs to provide power to the system. These registers might need to be programmed even when there's a power role swap from source to sink and during the change in PDOs.

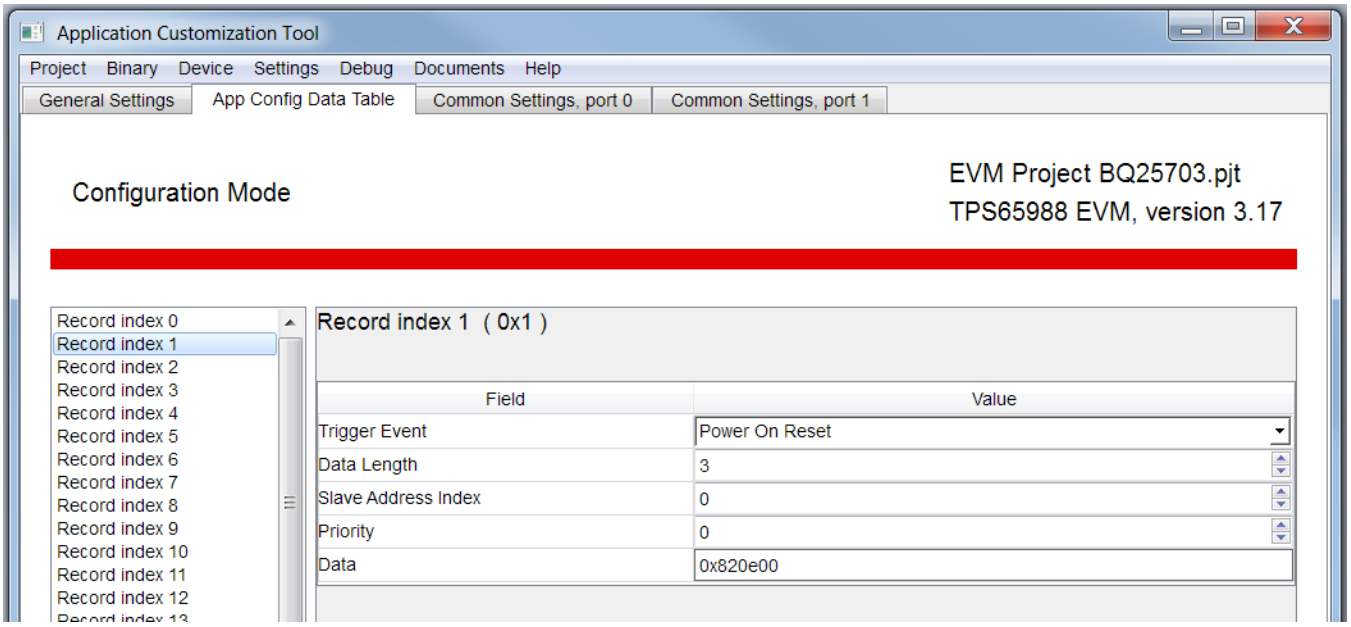
Register 0x35 needs to be programmed to enable OTG mode, apart from this a GPIO is also required to configure the BQ25703A as a power source or sink.

Remaining registers needs to be left at their default settings.

As discussed above these are the steps we need to take to configure the I2C master for BQ25703A:

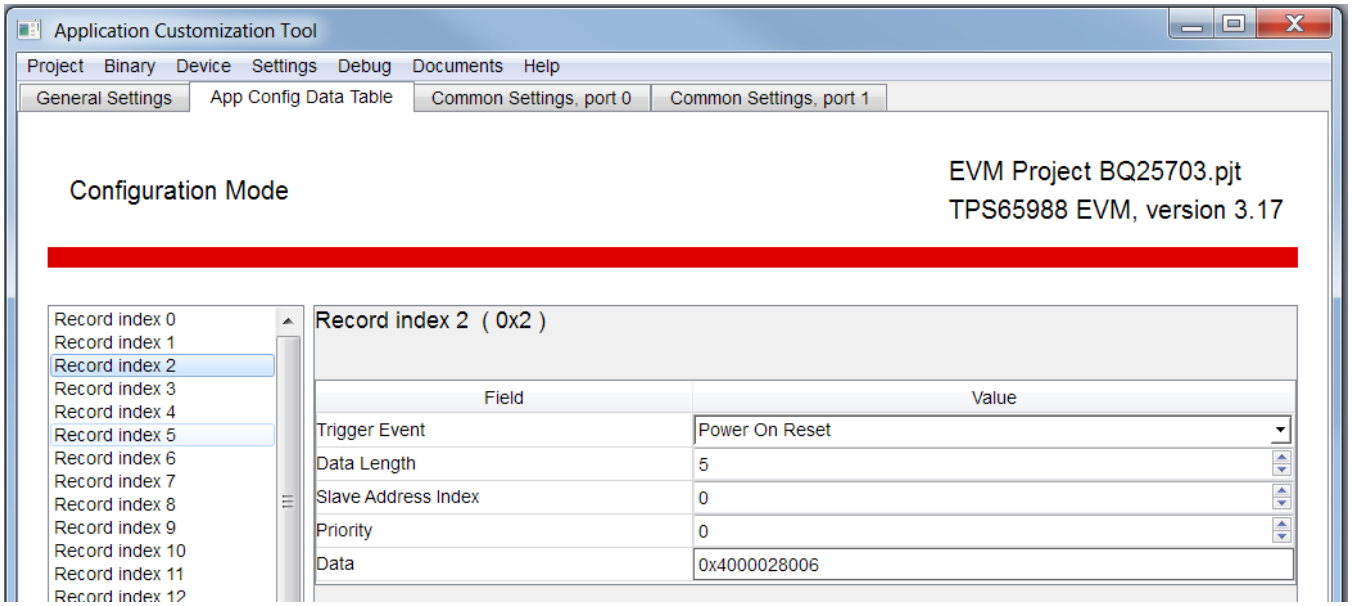
1. BQ25703A's registers 0x00, and 0x01 needs to be programmed with 0x0E and 0x82 respectively. To do this fill the record index 1 with 0x820E00. This data means the register address 0x00 will be written with 0x0E and 0x01 will be written with 0x82.

Figure 6. Record Index 1 for BQ25703A



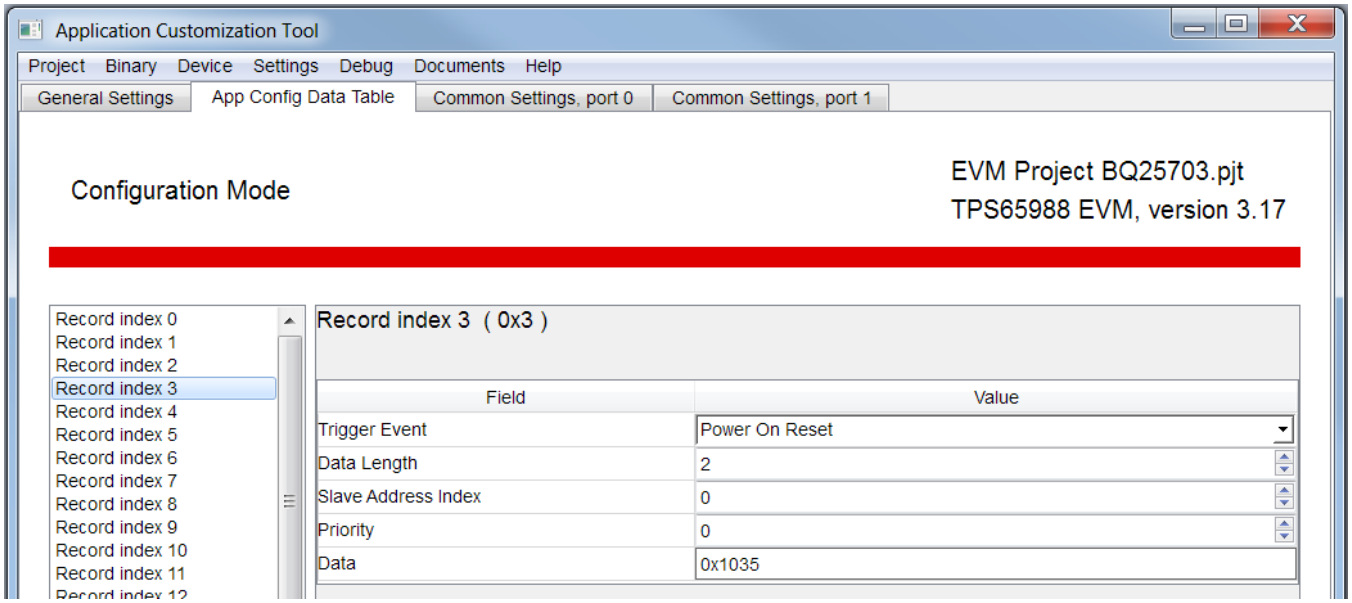
2. Now the OTG Voltage and OTG Current register needs to be set. Since in a PD source everything starts with a 5V/3A set the OTG Voltage to 5 V and OTG Current to 3 A. The OTG Current register can be programmed to provide 5 A current for application requiring up to 20V/5A PDOs, as the current can be controlled by the PD controller and therefore the BQ25703A can be set to the highest current. Write 0x80, 0x02, 0x00 and 0x40 to registers 0x06, 0x07, 0x08 and 0x09 respectively. These values will set the OTG Voltage to 5.12 V and OTG Current to 3.2 A.

Figure 7. Record Index 2 for BQ25703A



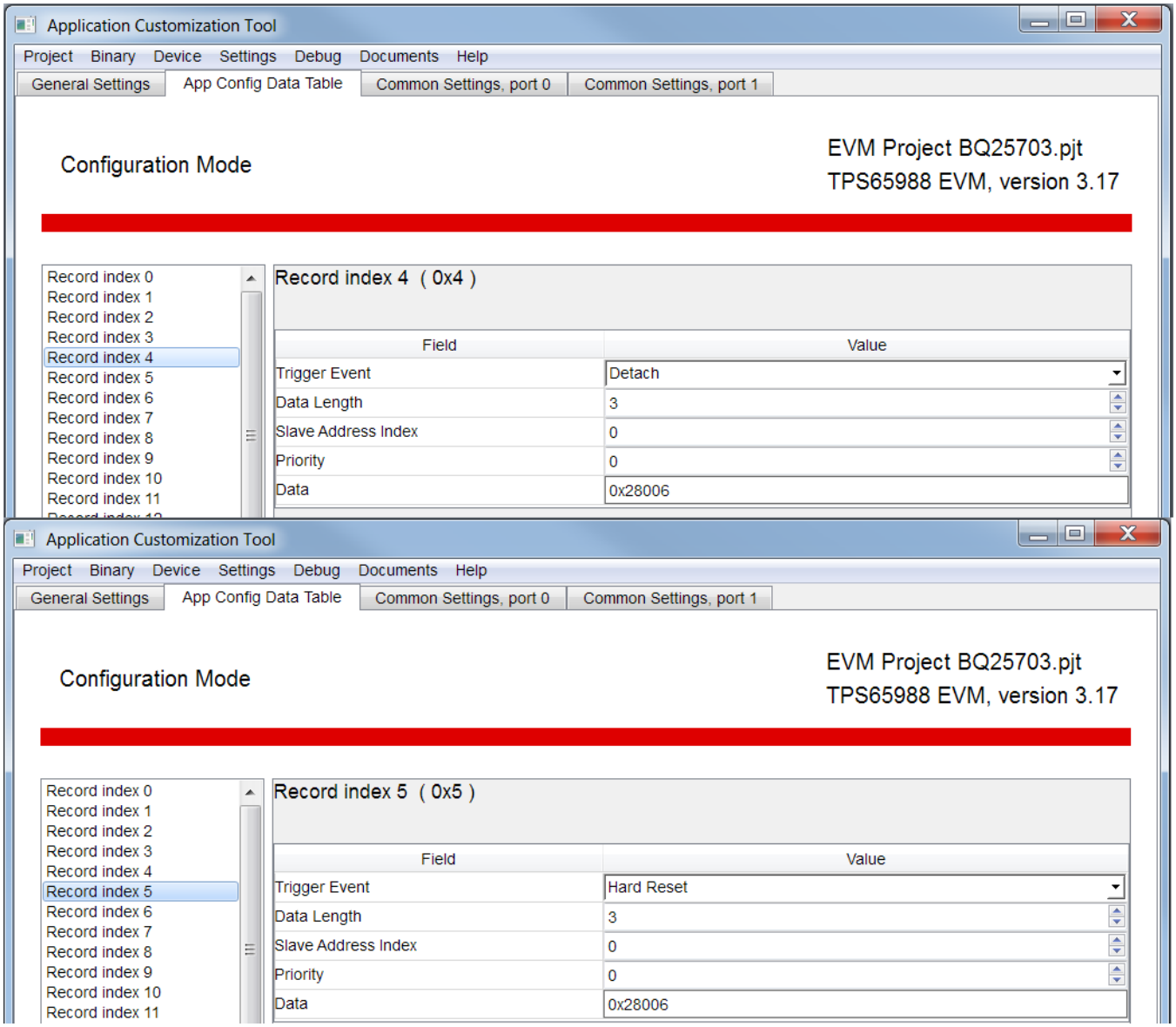
3. Now OTG Mode needs to be enabled by writing 0x10 to register 0x35.

Figure 8. Record Index 3 for BQ25703A



4. In case of Hard Reset or Cable Detach the output of Bq25703A needs to be brought back to 5 V and therefore we have to write 0x80 and 0x02 to registers 0x06 and 0x07 respectively.

Figure 9. Configuration for Hard Reset and Detach events



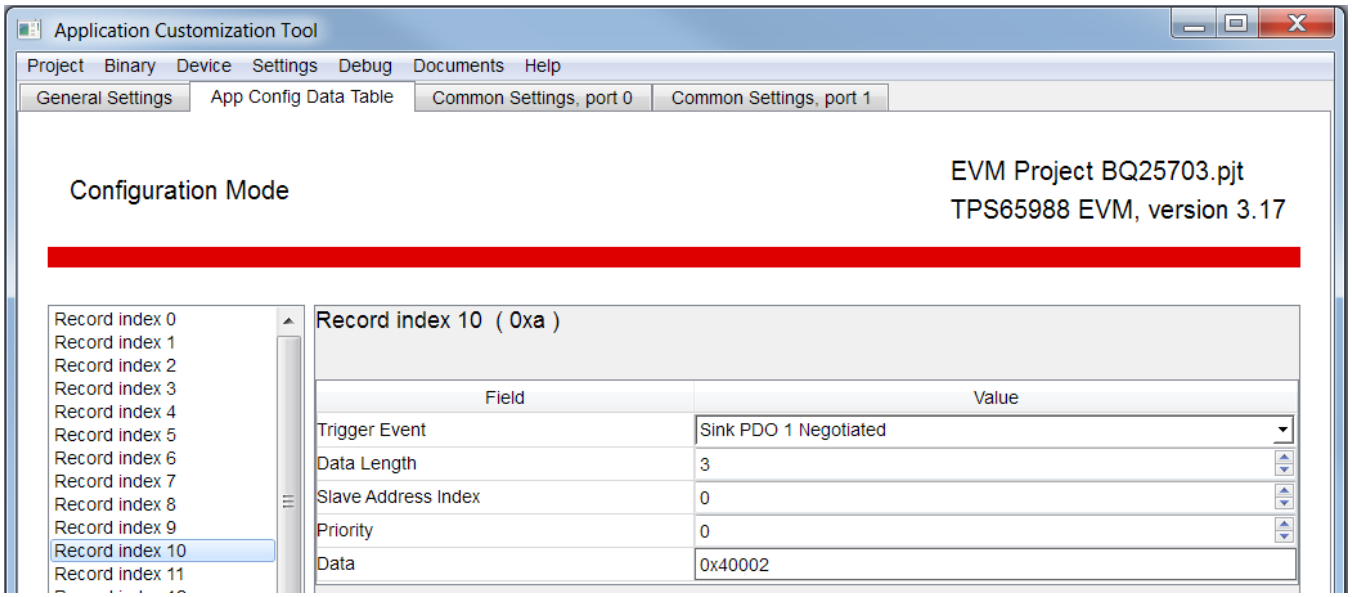
5. The registers 0x06 and 0x07 needs to be updated with relevant value whenever a Source PDO is negotiated. Please note the register needs to be programmed with a slightly higher values to compensate for the drops. The values for 5 V, 9 V, 15 V and 20 V PDOs are as follows:

PDO	Data
5 V	0x028006
9 V	0x130006
15 V	0x2B0006
20 V	0x3F0006

6. Similarly fill the remaining Record Indexes corresponding to other Source PDOs.
7. The registers 0x02 and 0x03 needs to be updated with relevant charging current whenever a Sink PDO is negotiated. Please note that this value tells the charging current of the Battery and therefore the ratio of battery voltage to the Sink PDO voltage needs to be factored in. For simplicity battery voltage can be taken as 12 V. The values for 5 V, 9 V, 15 V and 20 V PDOs are as follows:

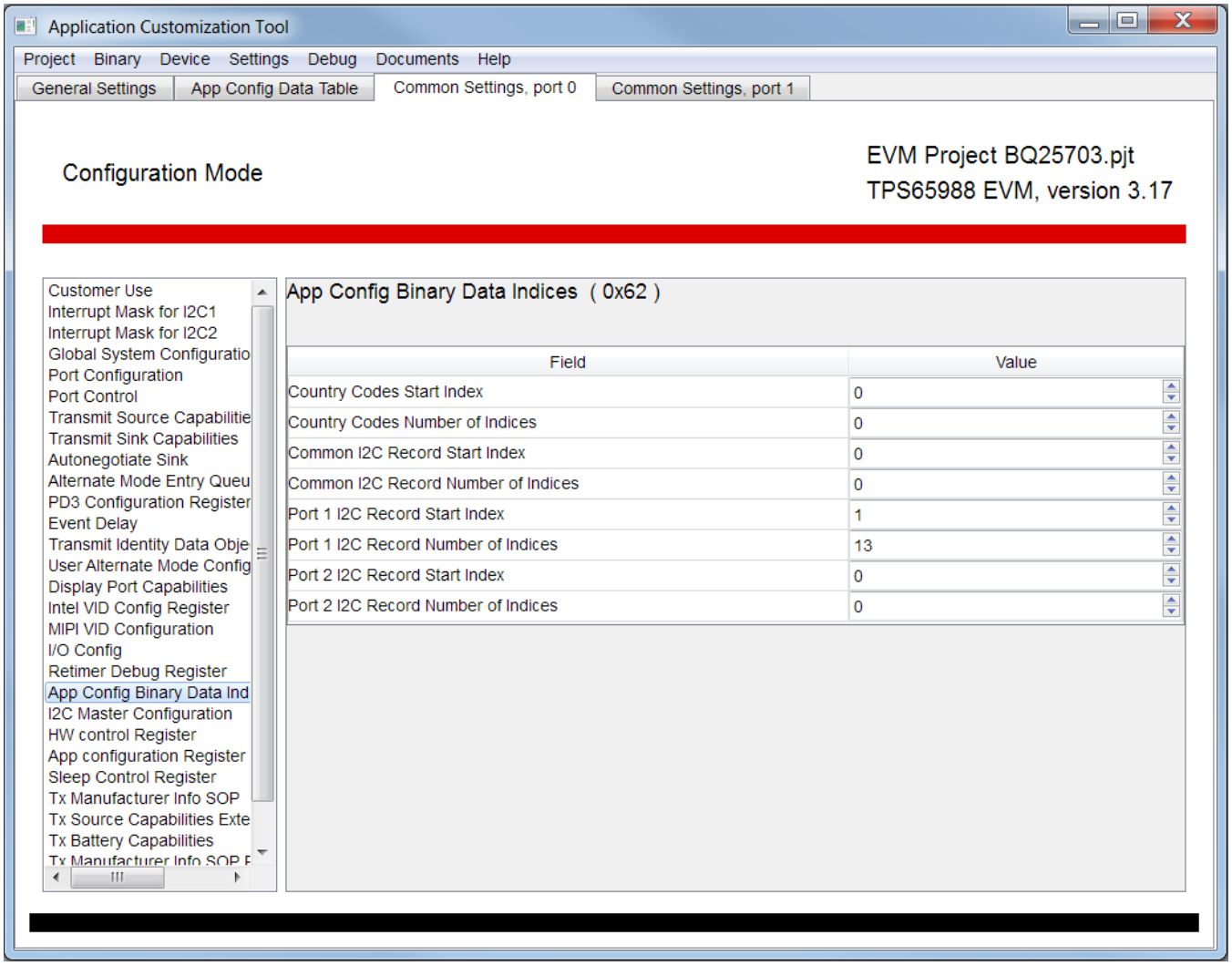
PDO	Total power	Charging current (Max 1.5 A)	Total power delivered to battery	Data
5V/3A	15 W	1 A	12 W	0x040002
9V/3A	27 W	1.5 A	18 W	0x060002
15V/3A	45 W	1.5 A	18 W	0x060002
20V/3A	60 W	1.5 A	18 W	0x060002

Figure 10. Record Index for 5 V Sink PDO



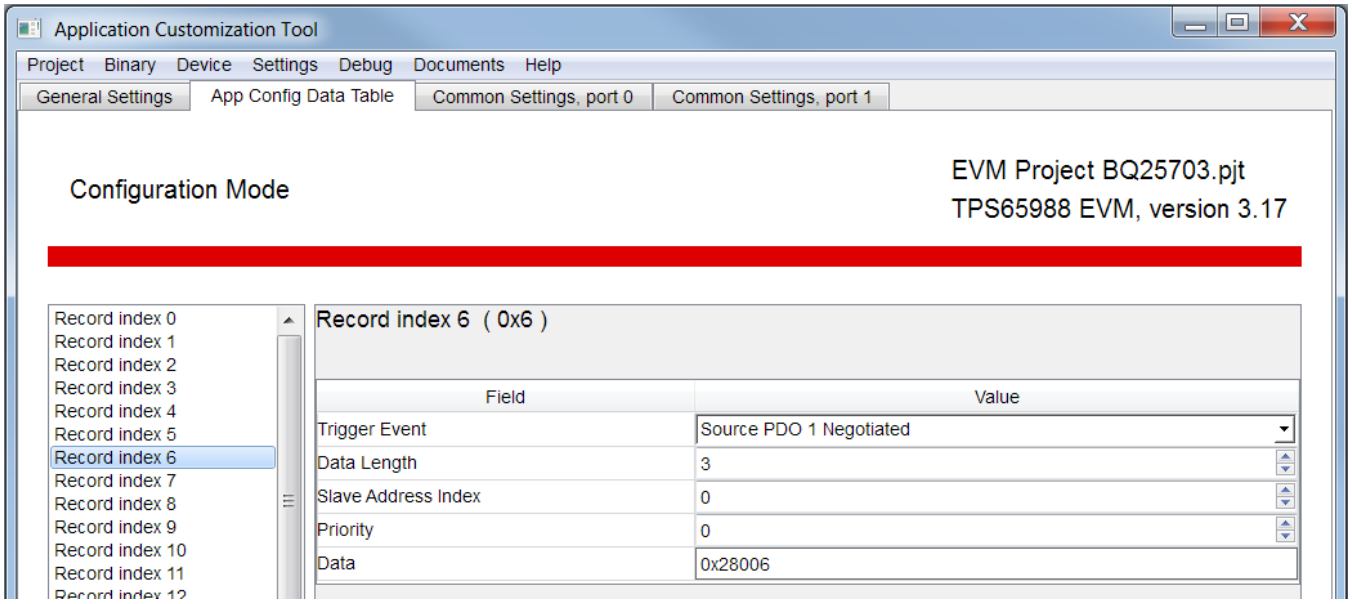
8. Similarly fill the remaining Record Indexes corresponding to other Sink PDOs.
9. Now go to App Configuration Binary Data Indices and fill it as below Image:

Figure 11. App Configuration Binary Data Indices for BQ25703A



10. Now you can create the firmware bin file, which can be used to program your system.

Figure 12. Record Index for 5 V Source PDO



5 Configuring I2C Master for TUSB546 Alternate Mode Linear Redriver Crosspoint Switch

The TUSB546 is a VESA USB Type-C alternate Mode redriving switch supporting USB 3.1 and DisplayPort 1.4 for hosts.

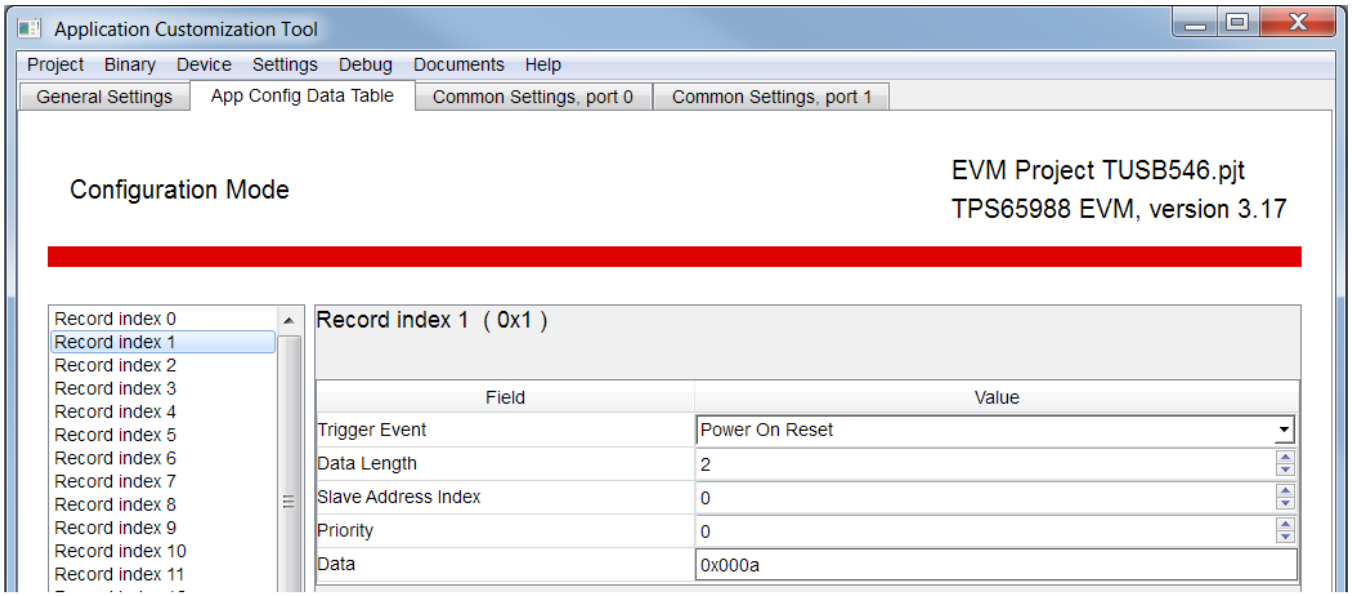
This device has 8 registers of 8 bits each, out of these 8 registers only 1 registers needs to be updated. For more details about the registers please refer the TUSB546 datasheet. Below table gives the details of various values which needs to be programmed to the register address 0x0A in different scenarios.

Mode	Cable Orientation	Reg Value
All Disabled	Straight	0x00
	Flipped	0x04
USB 3.1 Only	Straight	0x01
	Flipped	0x05
Four Lane DisplayPort	Straight	0x02
	Flipped	0x06
Two Lane DisplayPort	Straight	0x03
	Flipped	0x07

Configure a project as discussed in Section 3 of this application report and once done program the Record Indices in App Configuration Data Table window as follows:

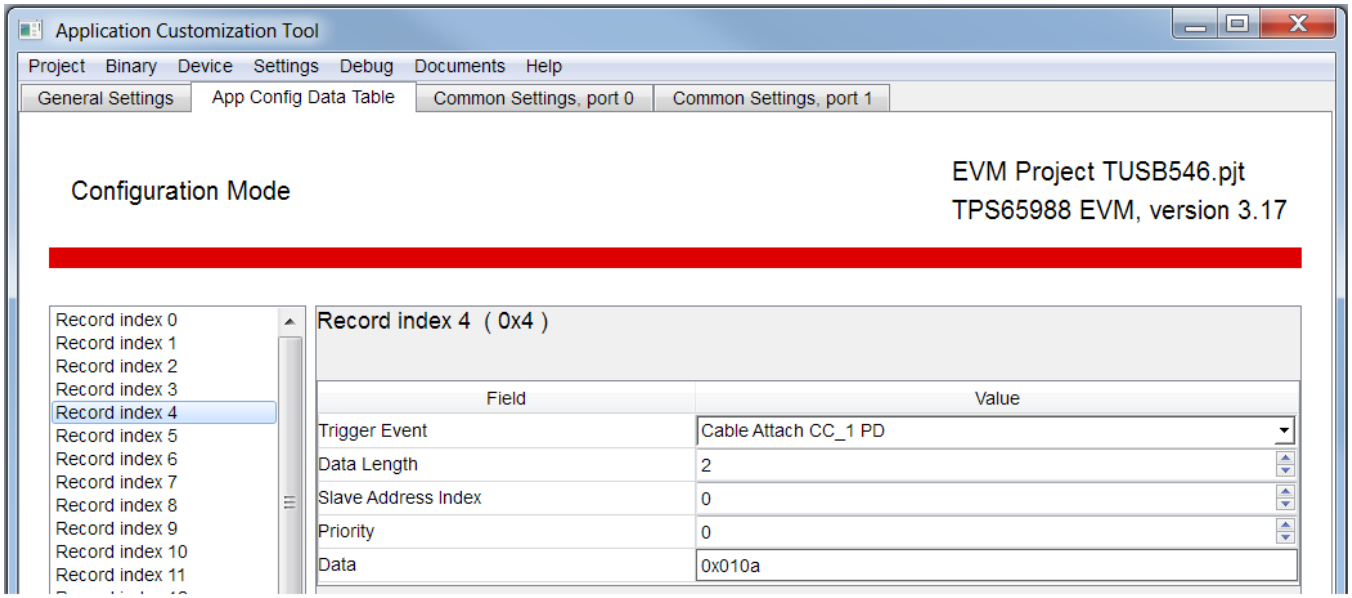
1. During Power On Reset, Detach and Hard Reset event the multiplexer should be disabled. Therefore data for the Record Indices corresponding to these events should have 0x000A.

Figure 13. Power Up Data for TUSB546



2. For Cable Attach CC_1 PD and DisplayPort Exoted CC_1 PD events data in the Record Indices should be 0x010A.

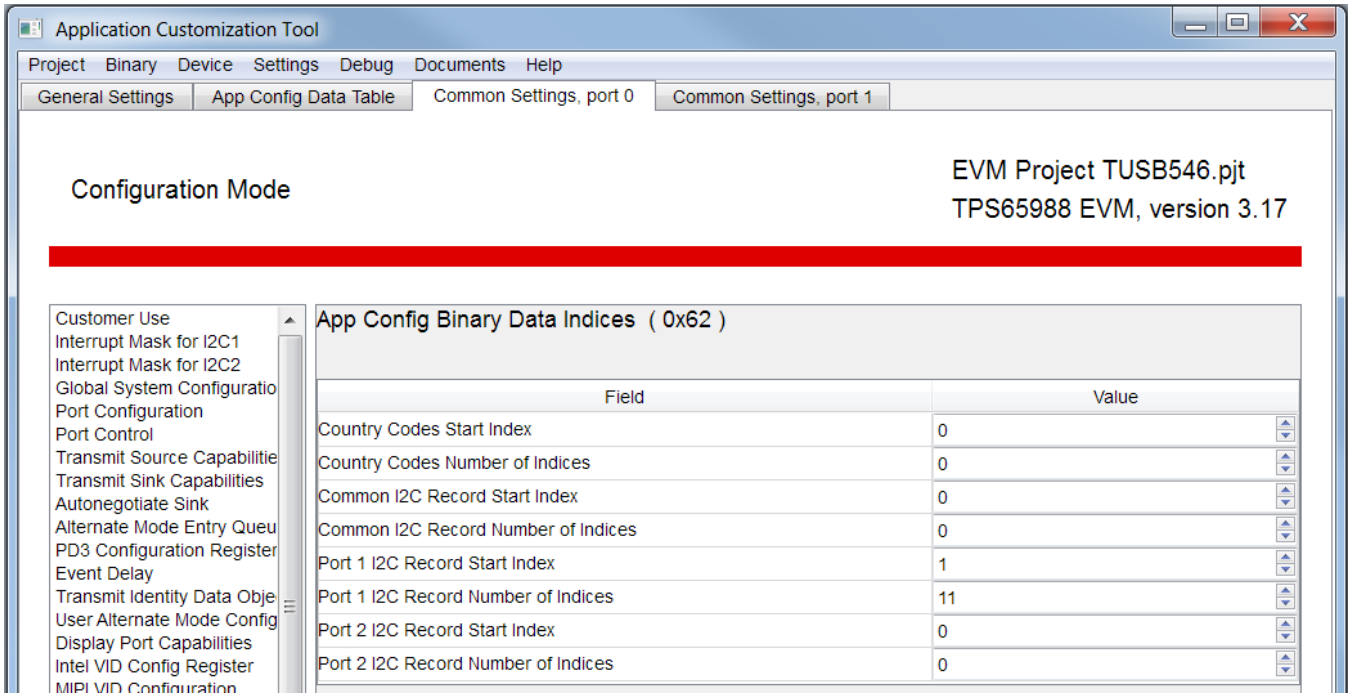
Figure 14. Configuration Data for TUSB546



3. For Cable Attach CC_2 PD and DisplayPort Exoted CC_2 PD events data in the Record Indices should be 0x050A
4. For DisplayPort Pin Configuration A, C or E CC_1 PD event the data sent to the mux should be 0x020A.
5. For DisplayPort Pin Configuration A, C or E CC_2 PD event the data sent to the mux should be 0x060A.

6. For DisplayPort Pin Configuration B, D or F CC_1 PD event the data sent to the mux should be 0x030A.
7. For DisplayPort Pin Configuration B, D or F CC_2 PD event the data sent to the mux should be 0x070A.
8. Fill the App Configuration Binary Data Indices as shown in below image:

Figure 15. App Config Binary Data Indices for TUSB546



9. Now you can create the firmware bin file, which can be used to program your system.

6 Summary

This Application Note explains how to configure the I2C Master for a given I2C slave and gives example of two of the most commonly used I2C slave in a PD ecosystem. User should be able to configure the I2C master for any arbitrary I2C slave by referring this application note.

7 References

- Texas Instruments, TUSB546-DCI USB Type-C™ DP ALT Mode Linear Redriver Crosspoint Switch data sheet
- Texas Instruments, bq25703A I2C Multi-Chemistry Battery Buck-Boost Charge Controller With System Power Monitor and Processor Hot Monitor data sheet

A.1

Below table list the events available in App Config Record Indices.

Event	Description
NULL	NULL event, should be used when no event is required for the record index
Cable Attach CC_x PD	Cable attached event on a particular CC line
Displayport Entered CC_x PD	Displayport entry event on a particular CC line
Displayport Pin Config A, C or E CC_x PD	4 lane Displayport entry event on a particular CC line
Displayport Pin Config B, D or F CC_x PD	2 lane Displayport entry event on a particular CC line
Thunderbolt Entered CC_x PD	Thunderbolt entry event on a particular CC line
User Alt Mode N Entered CC_x PD	Nth User Alt Mode entry event on a particular CC line
Debug Device Attached CC_x PD	Debug Device attached to a particular CC line
Audio Device Attached CC_x PD	Audio Device attached to a particular CC line
Displayport Exited CC_x PD	Displayport exit event on a particular CC line
Thunderbolt Exited CC_x PD	Thunderbolt exit event on a particular CC line
User Alt Mode N Exited CC_x PD	Nth User Alt Mode exit event on a particular CC line
Power On Reset	This event is triggered during the boot sequence of the PD controller
Detach	Detach event is triggered when a cable detach is detected
Hard Reset	This event is triggered when HARD RESET event is detected
Source PDO y negotiated	This event is triggered when yth Source PDO is selected
Sink PDO y negotiated	This event is triggered when yth Sink PDO is selected
Billboard Enable	This event is triggered when Billboard needs to be enabled
Billboard Disable	This event is triggered when Billboard needs to be disabled
HPD High	This event is triggered when HPD line goes high
HPD Low	This event is triggered when HPD line goes low
HPD IRQ	This event is triggered when HPD Interrupt comes
VCONN On	This event is triggered when VCONN turns on
VCONN Off	This event is triggered when VCONN turns off
PP k Switch On	This event is triggered when kth PP switch is turned on
PP k Switch Off	This event is triggered when kth PP switch is turned off

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale (www.ti.com/legal/termsofsale.html) or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265