# Building a USB PD sink for single cell applications.

Abstract

Developing a USB Power Delivery (PD) sink solution with high flexibility using the MAX77958 USB PD controller and the MAX32660 low-power microcontroller offers a robust and efficient platform for modern power management applications. This guide aims to help developers build a dual role port (DRP) single-cell USB power delivery application, leveraging the features and capabilities of the MAX77958 and MAX32660. The integration of these components enables efficient power negotiation, control, and management in both sink and source roles.

Introduction

USB Type-C has become a universal standard for connectivity and charging, supporting both legacy BC1.2 adapters and modern USB PD 3.0 adapters. BC1.2 includes port types like Standard Downstream Port (SDP), Charging Downstream Port (CDP), and Dedicated Charging Port (DCP). USB PD 3.0 can deliver up to 100W (20V/5A) for various devices. When developing a USB Type-C charging application, it is essential to incorporate both BC1.2 and USB PD detection for backward compatibility, ensuring support for both legacy and PD adapters.

## USB Type C Application using MAX77958:

A group of rectangular blue rectangles with black text

Description automatically generated

*Figure 2. A Typical USB PD DRP solution.*

A typical USB power application includes a USB power adapter (source) and a USB power sink. The adapter comprises a USB PD controller and a DC-DC/AC-DC converter, while the sink includes a USB PD controller, battery charger, and battery. A microcontroller can be added for high flexibility applications. The USB PD controller's role depends on whether it is on the source or sink side, following the USB PD3.0 standard for negotiation between source and sink. The MAX77958 integrates the PD3.0 protocol, supporting both source and sink roles. On the sink side, the MAX77958 supports voltages from 5V to 20V, depending on the source's available power. It can be pre-configured or dynamically configured to select available PDOs based on battery configuration and optimal charger performance. When pre-configured, it can independently set charging current and negotiate for maximum power levels. USB Type-C charging applications are divided into two categories based on power requirements: standard USB Type-C charging for up to 15W and USB PD 3.0 for applications up to 100W.

## **USB PD Negotiation Sequence**

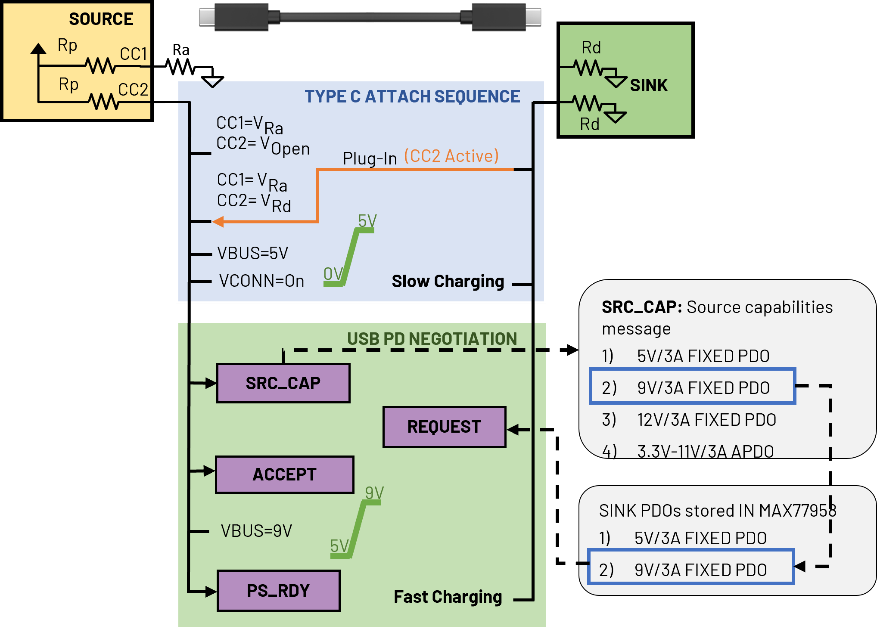
******With advances in consumer applications, power requirements are steadily increasing. To meet the demand for faster charging times, USB introduced the USB PD 3.0 standard, significantly increasing charging power capabilities.

Figure 2: USB PD Negotiation sequence

1. Figure 2 illustrates a typical power negotiation sequence when a USB PD compliant charger is connected to a sink with a PD controller like the MAX77958. Users can set desired sink PDOs using either a customization script configured by a GUI or an MCU script to configure the MAX77958’s non-volatile memory (MTP). In this guide, the MAX32660 is used to configure the Sink PDO settings of the MAX77958, which will be detailed in the following sections.
2. Once a connection is established between the source and the sink, the following Type-C detection sequence occurs:
3. Based on the cable's orientation, either CC1 or CC2 will be active. For instance, if CC1 of the source is connected to the sink's pull-down resistor (Rd), CC2 of the source will be connected to a Ra resistor and power the cable using the VCONN supply from the source.
4. The MAX77958, operating in Dual Role Port (DRP) mode, automatically switches to sink mode upon detecting a dedicated source on the active CC line.
5. The source pulls the VBUS to 5V, initiating slow charging on the sink side. The VCONN source is also turned on to power the cable, marking the end of the initial Type-C attachment sequence.
6. The source delivers its capabilities in the form of Power Data Objects (PDOs).
7. The MAX77958, configured with two sink PDOs, checks if the received source PDOs match its sink PDOs.
8. The MAX77958 selects the highest common PDO (e.g., 9V/3A) that the source can support and requests the source.
9. The source accepts the request, raises the VBUS to the required voltage level, and sends the PS\_RDY message to notify the sink.

This sequence can be easily performed by the MAX77958 in standalone mode by configuring its MTP once using standard MAX77958 commands sent from an MCU. These commands will be discussed in the next section.

***Type C Initial Plug-in PD negotiation*:**

In USB Power Delivery (USB PD), sink devices are classified into different types based on their power requirements, each corresponding to specific Power Delivery (PD) profiles. These profiles dictate the maximum power that a sink device can draw from a connected power source. For example, mobile phones and tablets typically require input voltages of 5V or 9V for efficient charging, while laptops demand higher voltages such as 20V. The MAX77958 is designed to configure various sink Power Delivery Objects (PDOs), accommodating different voltage and current levels as per the connected device's requirements. During connection to a power source, the MAX77958 engages in negotiations to determine the highest available power profile from the source. This negotiation process ensures optimal charging efficiency by aligning the source's capabilities with the sink PDO profiles. For mobile phone charging applications, which typically operate at either 5V or 9V with a current rating of 3A, the MAX77958 is configured with two sink PDOs: 5V/3A and 9V/3A. These PDOs are defined to align precisely with the voltage and current requirements of mobile devices, facilitating efficient and rapid charging. The configuration process involves setting these PDOs once, which will be stored in the MAX77958's MTP (One-Time Programmable Memory) by a microcontroller. This setup ensures that the MAX77958 autonomously selects the appropriate PDO based on the connected power source, enabling seamless and effective power delivery for mobile devices.

How to set the sink PDOs?

**Table 1. Sink PDO Format**

|  |  |
| --- | --- |
| Bit(s) | Description |
| B31…30 | Fixed supply |
| B29 | Dual-Role Power |
| B28 | Higher Capability |
| B27 | Unconstrained Power |
| B26 | USB Communications Capable |
| B25 | Dual-Role Data |
| B24…23 | Fast Role Swap required USB Type C Current  00b: Fast Swap not supported (default)  01b: Default USB Power  10b: 1.5A @ 5V  11b: 3.0A @ 5V |
| B22…20 | ***Reserved – Shall*** be set to zero. |
| B19…10 | Voltage in 50mV units |
| B9…0 | Maximum Current in 10mA units |

Consider we want to set two sink PDOs 5V/3A and 9V/3A. The hex value for 3A can be calculated by dividing 3000mA by 10mA, resulting in 300 (decimal) or 12C (hexadecimal), which constitutes bits [B9: B0]. Similarly, the register value for 5V is determined by dividing 5000mV by 50mV, yielding 100 (decimal) or 64 (hexadecimal), which constitutes bits [B19: B10]

A screenshot of a computer

Description automatically generated Combining these hex values, as shown in the figure below, into a 4-byte format, the Sink PDO1 is configured as 0x0001912C.

A screenshot of a computer

Description automatically generatedSimilarly, for the 9V/3A PDO, the register values are computed, and the corresponding 4-byte format for Sink PDO2 is established as 0x000B412C.

Using the "SNK PDO Set" command, the MAX32660 can configure Sink PDOs on the MAX77958. The user can configure up to 6 sink PDOs. Upon establishing a connection with a power source, the MAX77958 autonomously negotiates for the 9V/3A PDO, provided the connected source supports this power level. In this scenario, a Samsung 45W charger, which supports 9V output, is used. The figure below illustrates the initial voltage negotiation between a mobile phone, incorporating the MAX77958, and the power adapter. Following the connection, the negotiated voltage transitions from 5V to 9V. The active CC pin indicates that the MAX77958 operates in Dual Role Port (DRP) mode before the connection and then switches to a dedicated mode after the connection is established.

A screen shot of a computer

Description automatically generated

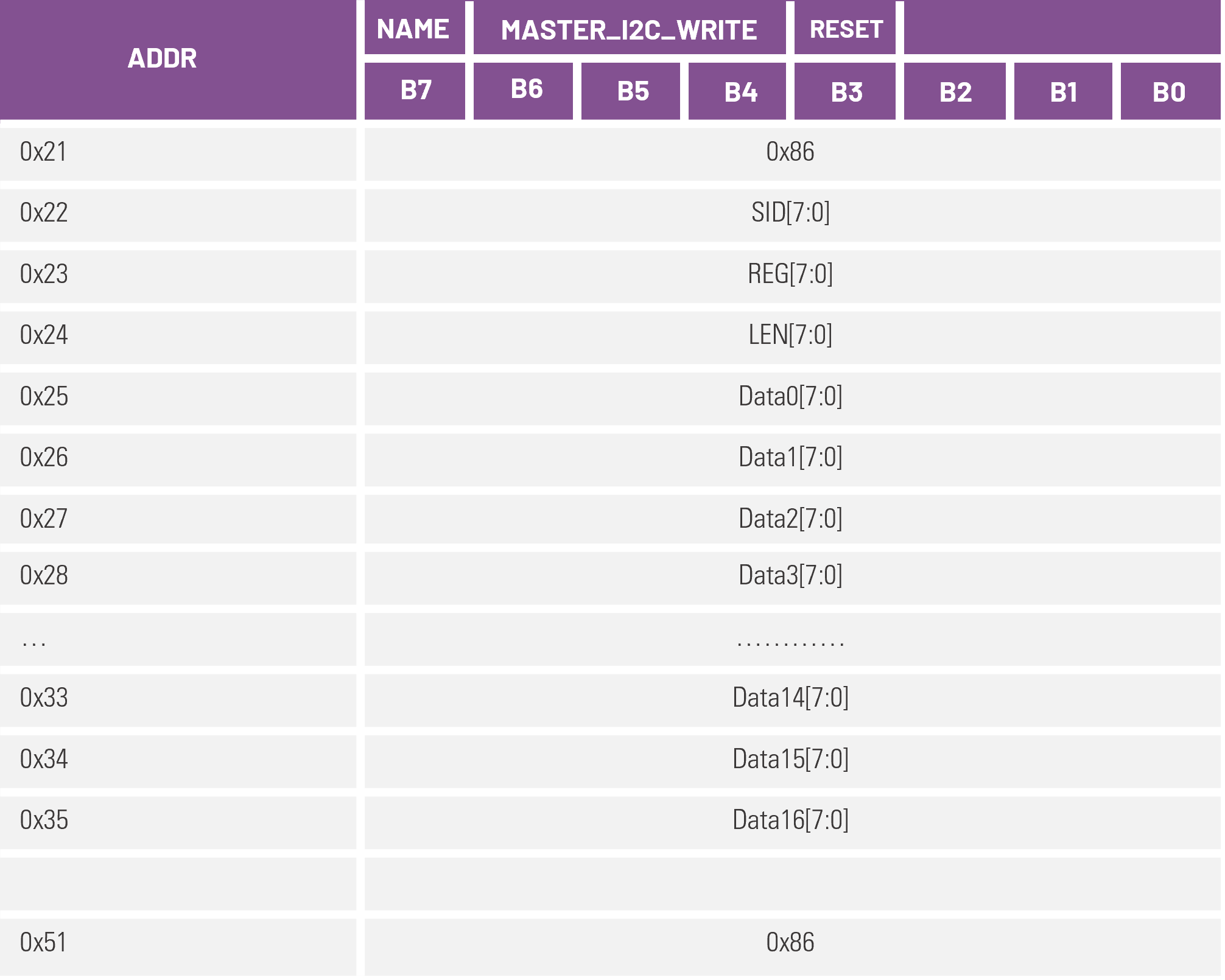
## **Integrating an MCU to the design**

## The MAX77958 can function in a standalone configuration for most Power Delivery (PD) applications needing sequential and event-based control. For applications requiring greater flexibility, integrating a microcontroller with the MAX77958 and the charger is beneficial. This setup simplifies complex functionalities and optimizes power management. It is ideal for robust and adaptable charging solutions, enabling features such as dynamic power negotiation, PPS charging for fast battery charging, dynamic current limiting for protection, and power management based on system requirements.

## **Using MAX77958 Master I2C port:**

## Before firmware development, it is better to understand how to use the MAX32660 to communicate indirectly with the MAX77986 via the MAX77958, as described in this section. This setup allows both the MAX77958 and MAX32660 to control the MAX77986, providing flexibility in applications where customization scripts and firmware are used together. Both MAX77958 and MAX32660 can act as masters to the MAX77986. For example, the MCU can dynamically program the charger's input current limit to reduce inrush current while the MAX77958 negotiates the required sink VBUS voltage during plugin. The MAX77958 features internal 32-byte buffers for transmitting and receiving standard opcode commands and data between the device and microcontroller. The two opcode commands for accessing the I2C master ports of the MAX77958 are the Master I2C Control Read command (0x85) and the Master I2C Control Write command (0x86).

**Table 3. Format for I2C master write.**

This section describes the indirect I2C register write method of MAX77986 using MAX32660 microcontroller via. the MAX77958. The operation is done by configuring MAX77958 “MASTER\_I2C\_WRITE” command and writing it to the write buffer of MAX77958. As an example, we will configure the input current limit of the charger by writing to CHG\_CNFG\_09 register (0x1F) of MAX77986. The table below shows the register map consisting of the buffers that must be written. The pseudo code below explains the sequence of writing.

Syntax: I2C\_write (slave addr, buff write addr, buff write data);

**Description**

**slave addr:** 7-bit I2C address of MAX77958 (0x25).

**buff write addr:** 8-bit address of the MAX77958 where the data needs to be written.

**buff read data:** 8-bit data that needs to be written.

**Pseudo code to write to MAX77986 register using MAX32660 via MAX77958:**

I2C\_write (0x25, 0x21, 0x86); // Opcode command for Master I2C write.

I2C\_write (0x25, 0x22, 0xD2); // MAX77986 I2C slave write address.

I2C\_write (0x25, 0x23, 0x1E); // MAX77986 register address to write.

I2C\_write (0x25, 0x24, 0x01); // Length of write data in byte.

I2C\_write (0x25, 0x25, 0x51); // Data written to defined MAX77986 register address.

(Fixing charger input current limit of MAX77986 to 2A)

I2C\_write (0x25, 0x41, 0x00); // Marks the end of buffer writes and process the command.

*For more details refer the MAX77958 API link attached in the last section*

*Please refer Opcode command guide for more information.*

## **USB PD CHARGING (>15W)**

## In recent years, most devices have adopted the Type-C connector with Power Delivery (PD). Compared to traditional BC1.2 and Standard Type-C charging, Power Delivery offers significantly higher power output, greatly reducing charging times. One of the primary limitations of traditional USB Type charging is the USB VBUS current limitation through the cables, with most standard cables supporting up to 3A. Therefore, for a standard VBUS voltage of 5V, the maximum power limit is capped at 15W. However, with the introduction of USB PD, the VBUS voltage can be raised up to 20V based on the negotiated contract between the source and sink devices. As a result, using standard cables that support up to 3A, the power output can increase to 60W. Additionally, with special E-Marked cables, the power can reach up to 100W.

**USB PD for Bi-Directional Charging using MCU.**

For applications like Power bank, the common Type C port be used for both charging the battery and providing power to external devices like mobile phone. In this scenario, the MAX77958 should be in this case works in a DRP mode with either Try.SNK on or off giving it flexibility to determine its role when another external DRP device is connected to the port. This section details how the MAX32660, MAX77958, and MAX77986 ICs support Power Delivery (PD) negotiation and bi-directional charging. Customization scripts in standalone mode or microcontroller implementation can manage this process. The flowchart below illustrates the PD charging sequence for the Dual Role Port (DRP) Type-C port of the MAX77958 under the influence of an MCU. When a PD power adapter (source) connects to the MAX77958 (sink), it generates an interrupt read by the MAX32660 via the INT pin. The MCU reads the MAX77958’s CCSTATUS0 register to identify the device type. If the device is a source, the MAX77958 acts as a sink. The MCU configures the input current limit of the MAX77986 using the CHG\_CNFG\_09 [6:0] register and enables charging mode by writing to CHG\_CNFG\_00. If the external device is a sink, the MAX77958 acts as a source. The MCU configures the OTG voltage and current by writing to CHG\_CNFG\_10 and CHG\_CNFG\_11 registers and enables reverse OTG mode by writing to CHG\_CNFG\_00. This marks the end of type C attach sequence.

A screenshot of a computer screen

Description automatically generated

*Figure: USB PD Bi-directional charging*

After the initial Type-C detection, the MAX77958 follows the USB PD sequence to negotiate for a power profile based on the available sink PDOs (Refer first 2 section for more details). The MCU can later prompt the MAX77958 to negotiate different power levels dynamically, covered in the "Dynamic Voltage Negotiation" section. If an external sink device is connected, the MAX77958 advertises its source PDOs. The sink requests a PDO, which the MAX77958 accepts. The MAX77986 is then configured with the OTG voltage and current limit, raising the VBUS to the requested power level.

**Dynamic Voltage Negotiation**

Voltage bus negotiation is essential for efficient power management in electronic systems, ensuring rapid and effective charging. In USB power delivery, the VBUS voltage can be dynamically negotiated based on the system's power requirements. For instance, in a setup where a charger like the MAX77986 is powering the system and charging the battery simultaneously, high system power consumption can prolong the battery charging time. To address this, the VBUS voltage can be renegotiated to a higher level, thereby increasing the available power to meet both system and battery charging needs. The MAX77958 employs Opcode commands to facilitate this dynamic voltage negotiation process. Two primary commands are used for this purpose: the "Get Src Cap" command and the "Src Cap Req" command. When the MAX77958 operates in sink mode, the "Get Src Cap" command retrieves the source capabilities in the form of Power Data Objects (PDOs). These PDOs contain crucial information such as voltage, current, and PDO type, and are stored sequentially in the buffer from positions 1 to 8. Once the sink device has the source capabilities, it can request a specific PDO using the "Src Cap Req" command. For example, in the USB PD bi-directional charging process, after the initial Type-C detection sequence, the MAX32660 microcontroller initiates the "Get Src Cap" command to the MAX77958. This prompts the MAX77958 to request the source capabilities from the power source. The source then responds by advertising its capabilities to the MAX77958. The microcontroller reads these source PDOs, selects an appropriate PDO, and configures the "Src Cap Req" command on the MAX77958 to request the selected PDO from the source.

The PDO is in a 4-byte format and can be realized to electrical quantities using the reference table provided below.

**Table 2. Source PDO Format**

|  |  |
| --- | --- |
| Bit(s) | Description |
| B31…30 | Fixed supply |
| B29 | Dual-Role Power |
| B28 | USB Suspend Supported |
| B27 | Unconstrained Power |
| B26 | USB Communications Capable |
| B25 | Dual-Role Data |
| B24 | Unchunked Extended Messages Supported |
| B23 | EPR Mode Capable |
| B22 | **Reserved – Shall** be set to zero. |
| B21…20 | Peak Current |
| B19…10 | Voltage in 50mV units |
| B9…0 | Maximum Current in 10mA units |

A screenshot of a computer

Description automatically generatedFor example, consider the PDO whose hex value is 0x0A01912C,

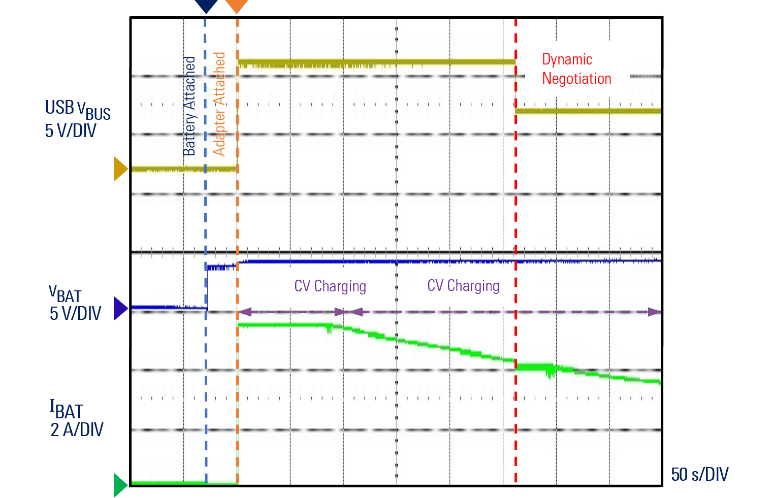
Maximum current = Source PDO [B9:B0] =   =

Using conversion factor of 10mA, Maximum current = 300 \* 10 = 3A.

Similarly, Maximum voltage = Source PDO [B19:B10] = =

Max voltage (in mV) = 100 \* 50 = 5V. Therefore, the PDO can deliver up to 15W (5V/3A)

To demonstrate dynamic voltage negotiation, consider a sink side setup as shown in Figure 1. The charging efficiency is mainly influenced by the switching charger MAX77986, positioned between the USB port and the battery. In a scenario where the battery requires a higher charging current to reduce charging time, the charger operates in constant-current charging mode. If the battery's power demand exceeds 15W, the only option to meet this demand is to renegotiate for a higher VBUS voltage. However, for a single-cell buck charger, higher input voltage results in lower efficiency for a given charging current. Therefore, it is recommended to maintain a lower bus voltage to optimize efficiency. In situations where the charger enters constant-voltage (CV) charging mode and the power requirement decreases, the VBUS voltage can be renegotiated to a lower level. This adjustment improves charging efficiency and heat management. The figure below illustrates this dynamic voltage negotiation, where the VBUS voltage is reduced from 9V to 5V during CV charging to enhance system efficiency. A highly accurate fuel gauge is employed to monitor the battery voltage and current. This data enables the microcontroller to instruct the MAX77958 to renegotiate for a lower VBUS voltage when the charging power requirement is 15W or below. Likewise, the voltage can be negotiated back to 9V or higher based on the power requirement for the system or the battery. For applications like power banks, the GPIOs of the MAX32660 can be configured to drive LEDs based on the battery's State of Charge (SoC).

****

**Q1. USB PD detection sequence**

**Q2. When to use buck boost and buck charger**

**Q3. What all are the requirement to implement sink side PPS when PPS source is connected?**

**Q3. Criteria for selecting USB voltage when PD charger is connected.**

1. **On the fly voltage negotiation for system efficiency (adding thermal images for 9V and 5V CV charging comparison)**

**e.g., 9V during CC and 5V during CV**

**Q4. How to configure USB sink to reverse OTG mode.**

**Q5. How to Display SOC using Fuel Gauge?**

**Q6. Role of MCU: Charge indication, On the fly OTG voltage**

**Q7. Changing OTG voltage and current based on Age of the battery (patent)**

**Q8. On the fly charging current limiting/USB input current limit to reduce inrush current.**

**Q9. Low power mode of MAX77986 using MCU. (Add some generic points on the system)**

**Q10. When SOC is below certain percentage then the microcontroller can change the port role from DRP to Sink mode on the fly.**