USB PD Demo Board User Guide

Table of Contents

Board Overview	2
Getting Started	3
Calibration Procedure	6
Charger Characteristics	7

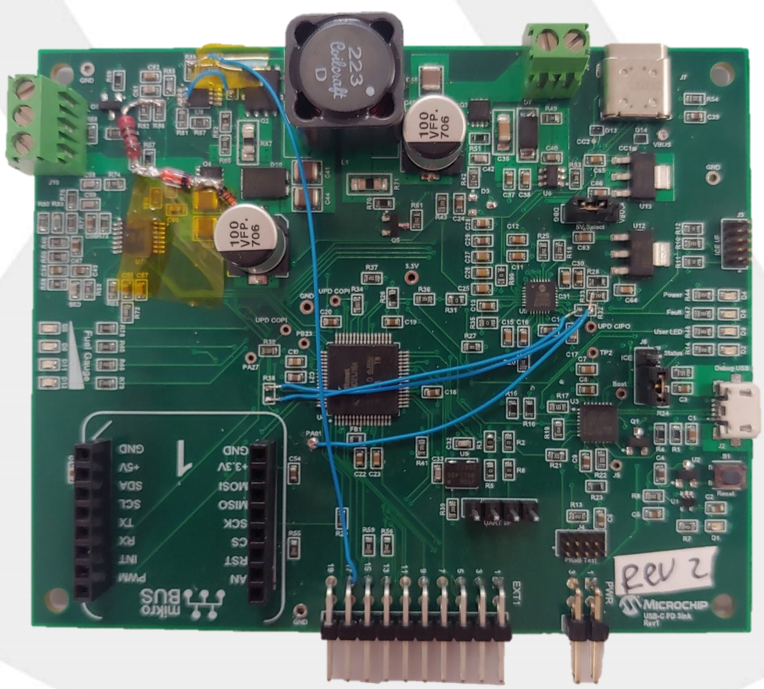


Figure 1, PD Demo Board

Board Overview

The USB PD Demo Board is a USB power delivery based battery charger demo board featuring the ATSAMD21J18A microcontroller. The board includes a PKoB for USB programming/debugging, along with an Atmel ICE interface.

Two types of expansion headers are supported by the board. There is one XPlained Pro I/O header along with a 4-pin XPlained Pro power header, and a mikroBUS click board connector.

The battery charger SEPIC power supply can support the full 20V/5A 100W USB PD specification.

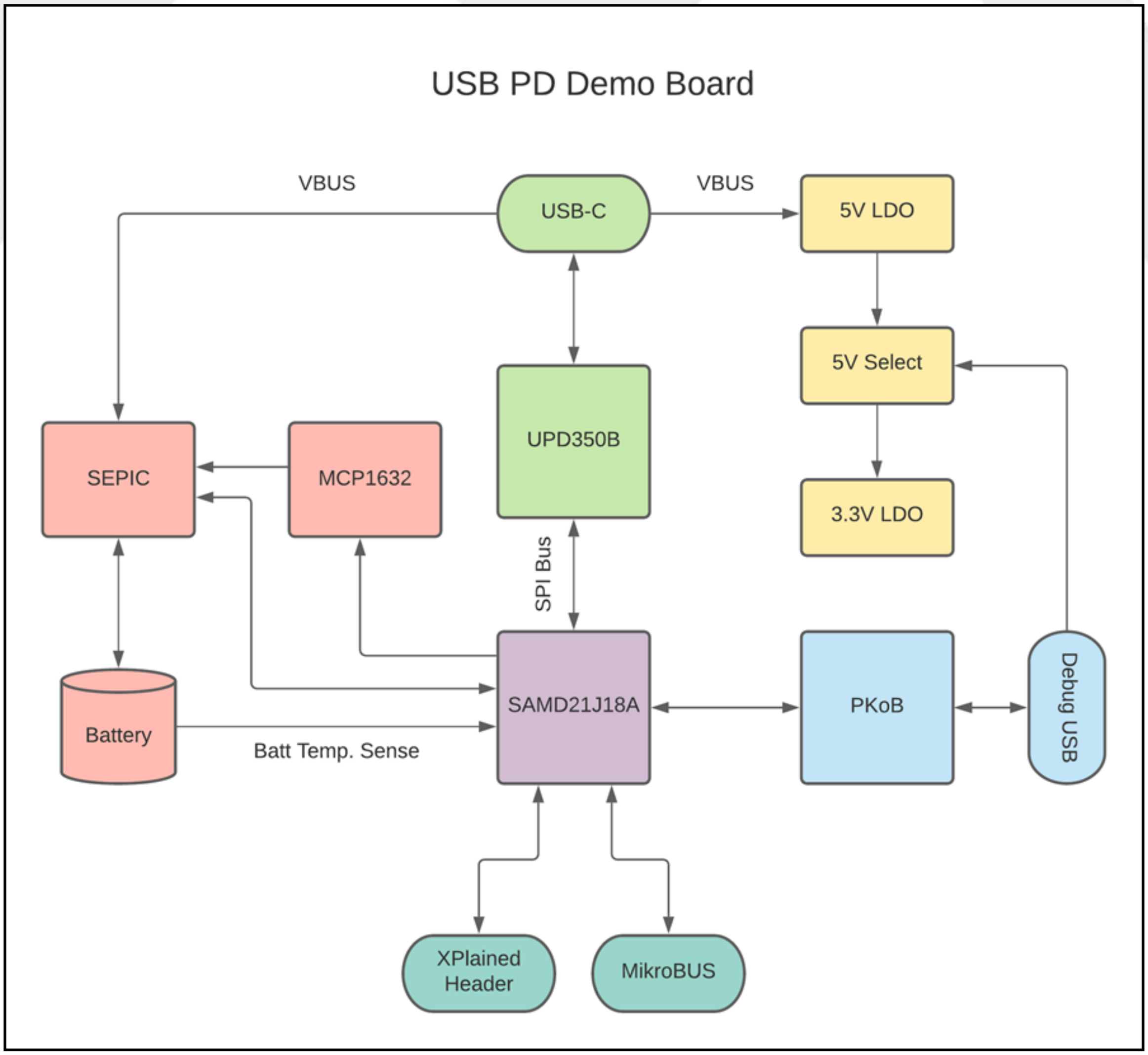


Figure 2, Block diagram

Getting Started

1. Download and launch MPLAB X IDE.
2. Plug the debug USB into your computer and make sure that MPLAB X recognizes a kit is attached. If the “Power” LED does not light up, make sure the 5V select header is set to the correct source.
3. Download and unzip the [demo code]
4. Open MPLAB X and click File > Open Project, then navigate to where you downloaded the file and go to the folder PSF_EVB_Sink—OLED1_working > PSF > Demo > PSF_EVB_Sink > firmware and select the project file
5. Build and program the demo code by pressing the  button located on the toolbar.
6. Connect the positive side of a 12V battery to the battery terminal marked “+” and the negative side of the battery to the terminal marked “-”.
7. Connect any USB PD capable charger to the USB-C connector to begin charging.

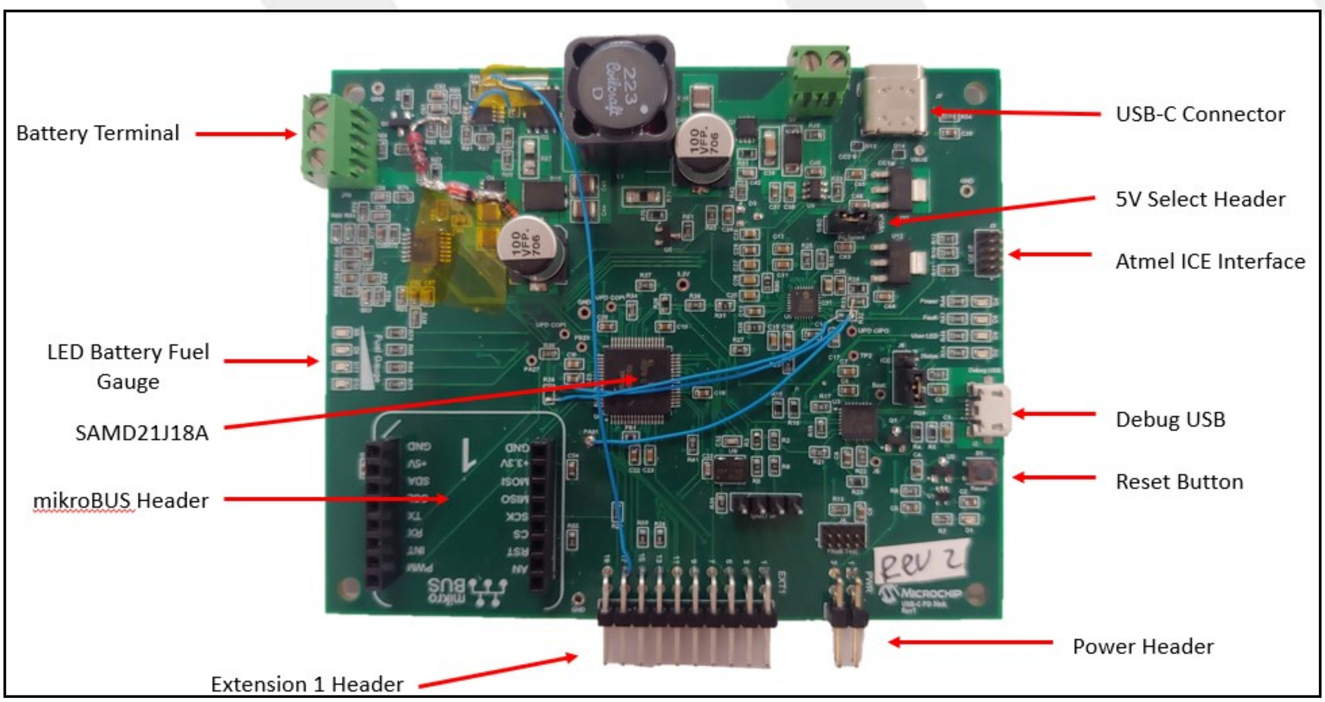


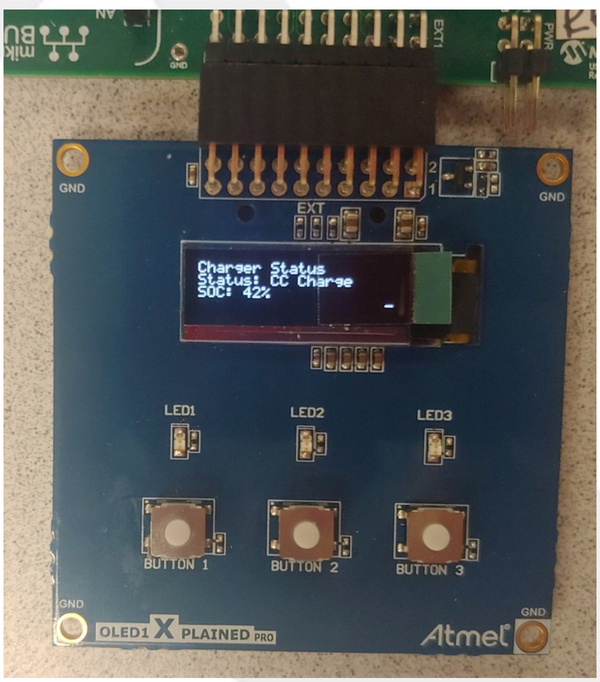
Figure 3, Board layout

Getting Started

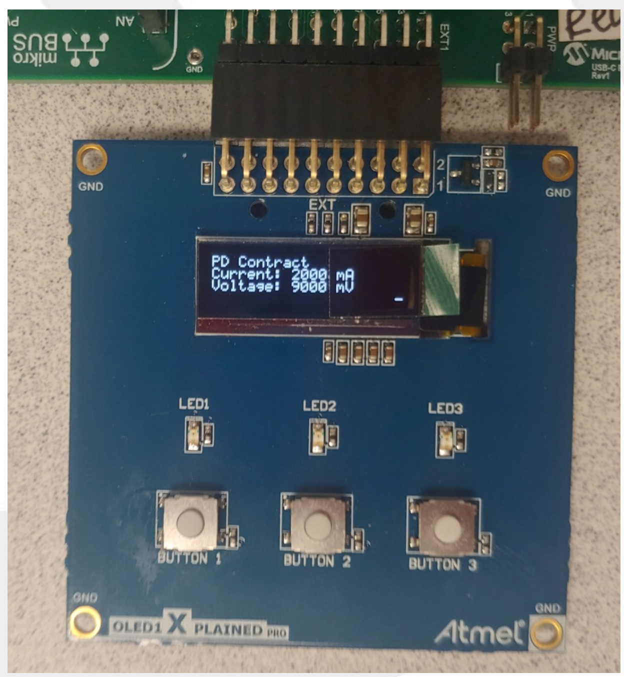
The demo code supports the OLED1 XPlained Pro add-on board on extension header 1. This add-on board is optional but is a useful tool for debugging and charger status monitoring.

Button 3 on the OLED1 board is used to switch between two display pages. On the first page, the battery charger status is shown (either Fault, Pre-condition, CC Mode, CV Mode, or Fully Charged). If a fault has occurred, it will display what type of fault it is. If there is no fault, it will display the battery SOC as a percentage. On page 2, the negotiated PD contract is displayed in terms of negotiated voltage and current.

Note: You may have to press the board reset button after plugging in the OLED1 board if the display does not work initially.



OLED1 Add-on Page 1



OLED1 Add-on Page 2

Figure 4, Different pages of OLED1 display board

Getting Started

Figures 5 and 6 detail the different state and fault codes that are recognized by the charger state machine and will be displayed on the OLED1 board. A brief description of what each code means is given.

Status Type	Integer Code	Description
FAULT	0	A fault has been detected
PRECONDITIONING	1	Battery voltage is too low for full current charging
CCMODE	2	Constant current charge mode
CVMODE	3	Constant voltage charge mode
CHARGED	4	Battery is fully charged
RECHARGE	5	Battery voltage has fallen since being charged

Figure 5, Charger state machine status codes

Fault Type	Integer Code	Description
GENERIC	0	Unknown fault
NOSOURCE	1	No PD source is attached
UVLO	2	Battery terminal voltage is too low
OVLO	3	Battery terminal voltage is too high
OVERTEMP	4	Battery temperature is too high
UNDERTEMP	5	Battery temperature is too low

Figure 6, Charger state machine fault codes

Calibration Procedure

An optional calibration procedure can be done to improve the accuracy of the charger current sense readings. A multimeter will be required for this process.

Steps to calibrate current readings:

1. Construct the circuit shown in the diagram below.
2. In the code file "SEPIC_CTRL.c" change the CALEN variable to 1 and reprogram the board to enable the calibration.

```
#define CALEN 1    //calibration mode enable, 0 = off, 1 = on
```

3. Plug a PD power source in to the USB-C connector (not depicted below).
4. Using the data visualizer in MPLAB X, enter the current (in mA) displayed on the multimeter. Doing this for two different values will enable us to calculate the necessary calibration parameters.
5. These values are stored in EEPROM and the calibration only needs to be done once. You will have to repeat the calibration if you reprogram the board.

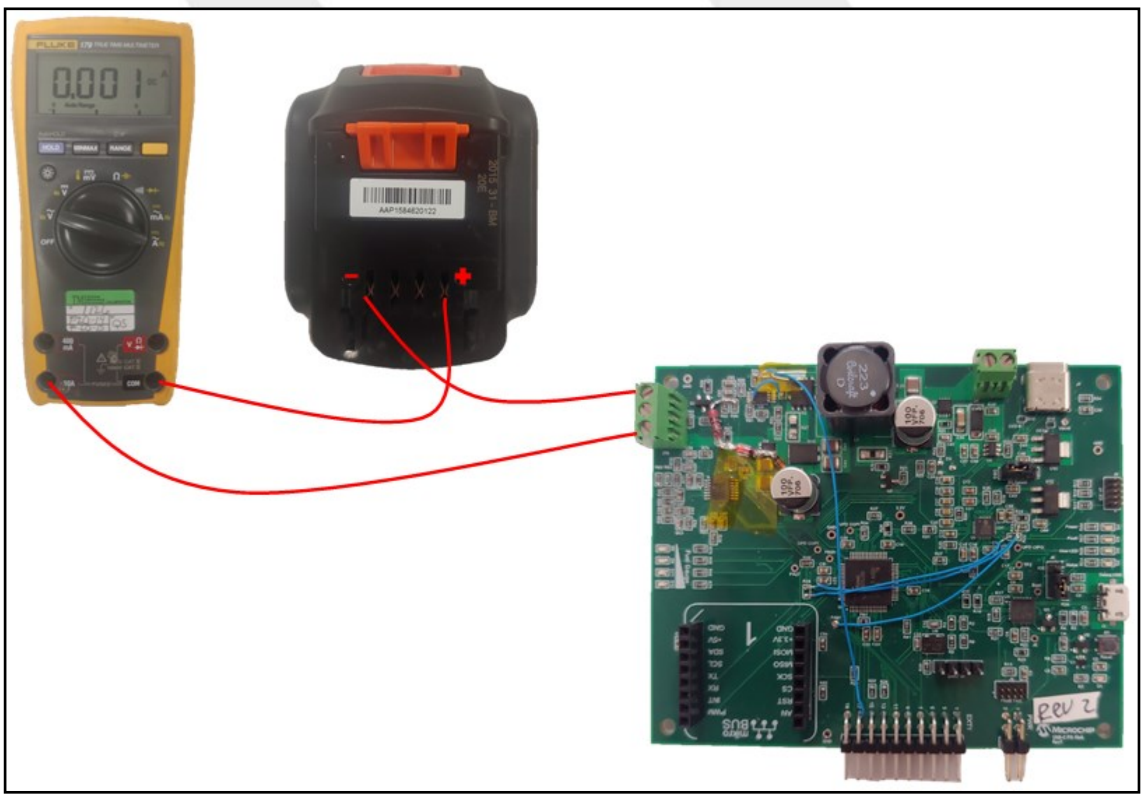


Figure 5, Calibration procedure diagram

Charger Characteristics

The charger uses a constant current/constant voltage charge algorithm. There are three main states that the charger operates in, pre-condition, constant current charge, and constant voltage charge. The charger will enter pre-condition mode if it detects the battery voltage is too low to safely charge at full current. In this mode, charge current is limited to a few hundred milliamps. Once the charger detects the battery voltage is above the pre-charge cutoff threshold, it will ramp up current to the maximum allowed charge current. This value can be hard-coded by the user or can be set to automatically calculate based on the negotiated PD contract.

The charger will continue to charge at constant current until the battery voltage nears its maximum voltage at which point it will enter constant voltage mode. In this mode, the charger checks the battery voltage every 500ms. If the voltage is above the maximum battery voltage, it will decrement current until it is at or slightly below that voltage threshold. This will maintain the battery voltage at a constant level. This process will continue until the charge current is below a specified cutoff current. At this point the charger will shutoff but will continue monitoring the battery and topping off the charge as needed.

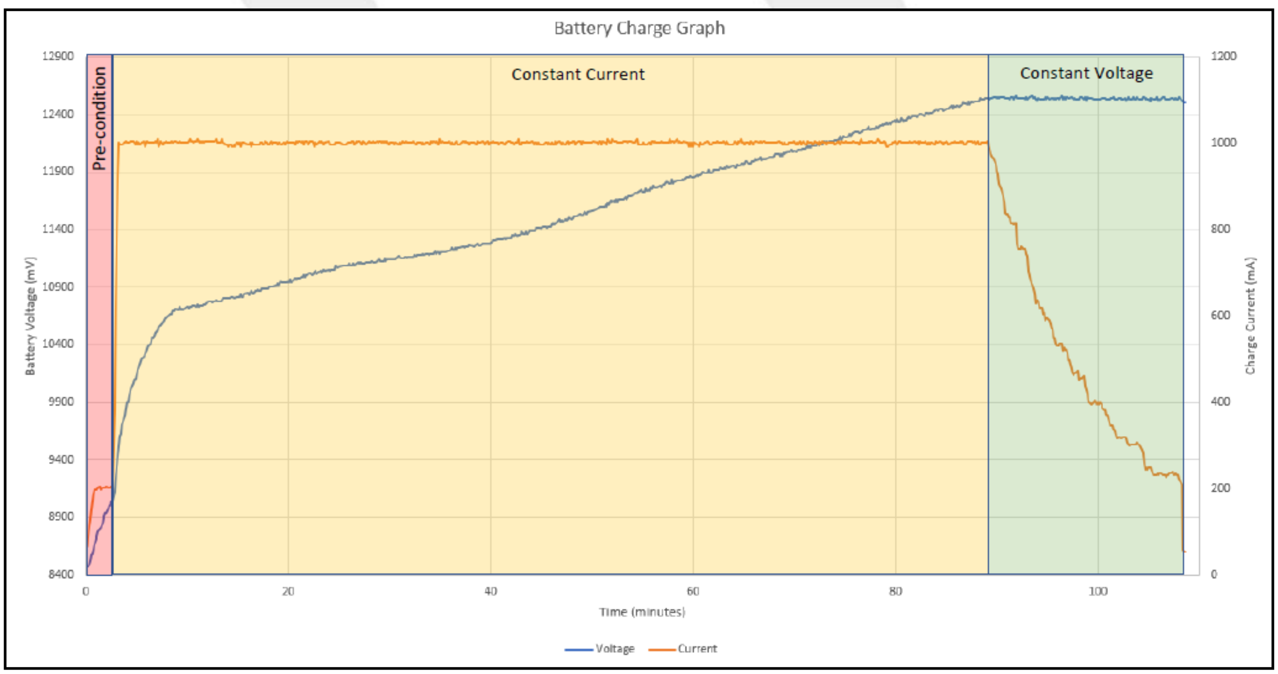


Figure 6, Charger characteristics graph

Charger Characteristics

The parameters for charger state thresholds can be tuned in the "SEPIC_CTRL.c" file. Several defines are used to establish battery parameters and desired thresholds/cutoffs.

```
//set battery parameters
#define CELLVMIN 2700 //inidivual cell min voltage in mV
#define CELLVMAX 4200 //inidivual cell max voltage in mV
#define BATTIMAX 3000 //max charge current in mA
#define NUMCELLS 3 //number of series cells
#define UCLO 150 //charge cutoff current in mA
#define BATTVMAX (NUMCELLS * CELLVMAX) //total battery maximum voltage
#define BATTVMIN (NUMCELLS * CELLVMIN) //total battery minimum voltage (UVLO value)
#define RECHARGETHRESH 4100*NUMCELLS //threshold for trickle charge engage
#define MINCCCHARGETHRESH 3000*NUMCELLS //threshold for full speed cc charging
#define CVTHRESH 4180*NUMCELLS //threshold to swtich from CC to CV charge
```

Additionally, the preferred charge current can be manually or automatically determined by modifying the code shown below.

```
    uint16_t maxcurrent = gasCfgStatusData.sPerPortData[0].u16NegoCurrentInmA;
    //set this value for a manual max charge current limit,
    //otherwise comment this line to use the PD negotiated current
    maxcurrent = 1000;
```