|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Microchip Technology Inc. | | | |  |
| REV | DATE | CN # | ORIGINATOR | DESCRIPTION OF CHANGE |
| 0.1 | 8/4/2020 |  | Riyas | Initial doc |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
| Electronic Distribution: | | | | |



PSF SINK Demo Design

Information contained in this publication regarding device applications and the like is provided only for your convenience and can be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE**.**

Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer’s risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights.

Table of Contents

[1 Introduction 5](#_Toc5883973)

[1.1 Scope 5](#_Toc5883974)

[1.2 References 5](#_Toc5883975)

[1.3 Compliance with Specifications 5](#_Toc5883976)

[1.4 Terms and Abbreviations 5](#_Toc5883977)

[2 Product Architecture Design 6](#_Toc5883978)

[2.1 Product Description 6](#_Toc5883979)

[2.2 Product Software Architecture 6](#_Toc5883980)

[2.3 Architectural Constraints 7](#_Toc5883981)

[2.4 Task Execution 7](#_Toc5883982)

[2.5 Memory Mapping and Constraints 8](#_Toc5883983)

[3 Boot Sequence 9](#_Toc5883984)

[4 Interfaces 10](#_Toc5883985)

[5 Source Code organisation 11](#_Toc5883986)

[5.1 Directory organisation 11](#_Toc5883987)

[5.2 Product Related Files 11](#_Toc5883988)

[6 Hardware and Software platforms 12](#_Toc5883989)

[6.1 Hardware 12](#_Toc5883990)

[6.2 Software 12](#_Toc5883991)

[7 Anticipated Changes 13](#_Toc5883992)

[8 Appendix 14](#_Toc5883993)

[8.1 MOM Links 14](#_Toc5883994)

[8.2 Review Checklist 14](#_Toc5883995)

[8.2.1 Version x.y 14](#_Toc5883996)

[8.2.2 Version x.z 14](#_Toc5883997)

# Introduction

This is the Design document for PSF SINK demo over the existing sample application provided by developer team.

## Scope

Scope of this document is the low level design details of PSF SINK demo firmware

The intended audience for this document is the firmware and application team.

## References

<List down the document references, if any referred in this document>

|  |  |
| --- | --- |
| **Document ID / File Name** | **Document Description** |
| PSF AE\_Sink requirements | Requirement document |
| PSF AE\_SINK\_DEMO | Demo proposal PPT |

## Compliance with Specifications

|  |  |
| --- | --- |
| **Specification** | **Spec Revision** |
| PD specification | 3.0 |
|  |  |
|  |  |

## Terms and Abbreviations

|  |  |
| --- | --- |
| **Abbreviation / Terminology** | **Description / Definition** |
| Module | Module is a software component to perform specific functionality |
| Interface | An interface is a shared boundary across which two or more separate components of a system exchange the information. |
| RTOS | Real-Time Operating System |
| Software Units | Software units are items that can’t be split into sub-items |

# Product Architecture Design

## Product Description

It is a PSF SINK application example that helps the customers to start evaluate their requirements with pre-built binaries.

## Product Software Architecture

ADC

Software Configurations

PSF SINK Library

Task Manager

LED

PSF Control Terminal

PDO Selector

UART

Harmony 3

## Architectural Constraints

<Specify the architectural design constraints if any.

## Reuse the existing configuration

Reuse the all the configuration parameter, application files s and Harmony project used in branch is labeled feature/upd301c-sink-ae-demo

## Demo folder structure

Use same folder structure as of used in branch is labeled feature/upd301c-sink-ae-demo

## Coding guidelines

Coding should strictly follow UNG coding guidelines

## PSF configuration

Reuse “PSFSink\_BootCfg.h”

Reuse “PSF\_Config.h”

Reuse all other .c file in application folder.

## PSF application

Reuse PSFSink\_BootCfg.c no need for further modifications

Reuse PSFSink\_App.c and modify it to add ADC Rotor and PCT.

## PSF Rotor switch

(Ref:- R\_FUNC\_ADC\_ROTOR\_SWITCH in Requirement doc)



The corresponding PDOs are,

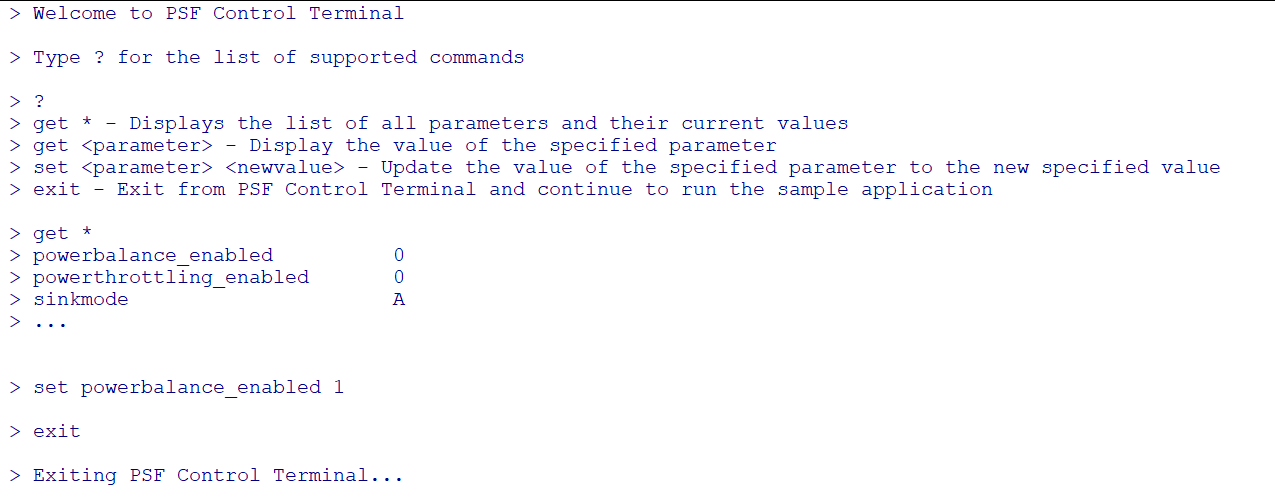
1. 5V @ 3A
2. 9V @ 3A
3. 15V @ 3A
4. 20V @ 3A
5. 20V @ 3A
6. 20V @ 3A
7. 20V @ 3A
8. Add ADC PLIIB using harmony configurator for PA04 of SAMD20
9. Poll for new ADC value in main while loop,
10. If it gets a new value,
    1. update “u32aNewPDO” as per the ADC position
    2. Set 5th bit of **u32ClientRequest**
    3. Set 10th bit of **u32CfgData**
    4. Wait for “eMCHP\_PSF\_PD\_CONTRACT\_NEGOTIATED” notification to make sure it is processed by PSF stack

## PSF control terminal

(Ref: R\_FUNC\_PFC\_CONTROL\_TERMINAL in Requirement doc)

Default baud rate – 115200

Welcome screen-



Commands to be implemented-

* **~~set br [value]~~** ~~- to set uart baud rate~~
* **set mem [memory\_address] [byte value]** - write memory of SAMD20
* **set pdo [position] [value]** – Insert a new PDO at the mentioned position
* **get version**
* **~~get br~~** ~~- Get baud rate~~
* **get pdo** - Current sink power details
* **get sr [Name]** - Read status registers
  + **Supported names**-
    - GlobalCfgStatusData
    - PortCfgStatus
* **get mem [memory\_address] [length]** - Read memory
* **Space bar - Enter Command prompt, and stop debug messages**
* **Space bar – Continue during bootup**
* **‘Q or q’ – Exit command prompt- continue debug messages if enabled**

1. Enable UART plib using HARMONY (It is already enabled if TRACE is enabled in PSF trace is enabled. Adda conditional macro to achieve both)
2. Use harmony PLIB USART read and write commands to implement set and get commands
3. If PCT is enabled, then code execution waits till continue (SPACE BAR) command to run main while loop. This helps the user to modify the predefined configuration values without recompiling the code
4. In main while loop, always monitor “SPACE BAR” for Enter or exit PCT.
5. If PCT is active, stop DEBUG TRACE messages, monitor for supported commands, and process it.
6. If PCT should not be a blocking task, if it needs to print 1000 characters, let it write 32 character in one cycle. In order to achieve this either implement interrupt write method or a separate write management for PCT. It will help to run the PSF stack smoothly
7. PCT is made case insensitive. It always converts input to lower case alphabets.
8. When using **set pdo** command, user must give the value of PDO as ((((voltage)/50) << 10) | ((current)/10)). For e.g. For the PDO (15000U,3000U), the pdo value to be given is 4B12C.
9. Program the EVB by setting the Rotor Knob position to position 4 (considering the positions start from 0) as that is the default position.

## PSF Current monitor

(Ref: R\_FUNC\_PFC\_CONTROL\_TERMINAL in Requirement doc)

1. Add ADC PLIIB using harmony configurator for PA05 of SAMD20
2. Monitor the value and print trace message periodically

## Memory Mapping and Constraints

# Boot Sequence

SYS\_Initialize (make sure it only added PCT, ADC for PA04 AND pa05)

MCHPPSF\_Init(timer, UART for trac, DAC for iDAC all already initialized here)

Wait for “Continue(SPACE)” PCT

MCHPPSF\_Run

Harmony tasks

Monitor and process ADCs(PA04 AND PA05)

Monitor and process PCT

# Interfaces

All the interfacing modules has been completed by design team itself, the config values are updated in requirement doc[Miscellaneous Requirements]

# Source Code organisation

## Directory organisation

# Hardware and Software platforms

## Hardware

AE-UPD301C\_BASIC\_SINK

## Software

1. IDE - Mplabx
2. Debugger- Atmel ICE
3. Coding Standard -
4. Version Control -
5. Bug Tracking tool -JIRA

# Anticipated Changes

Complete PCT implementation may not be available.

# Appendix

## MOM Links

<List down all the links of the meeting minutes document taken during the product design review>

## Review Checklist

### Version x.y

|  |  |  |
| --- | --- | --- |
| **Checkpoint** | **Response** | **More Information** |
|  |  |  |
|  |  |  |

### Version x.z

|  |  |  |
| --- | --- | --- |
| **Checkpoint** | **Response** | **More Information** |
|  |  |  |
|  |  |  |