Portland State
UNIVERSITY

Erebus Labs

STEM SENSORS

DESIGN SPECIFICATION

Version 1.0

1/28/2014

SCOTT LAWSON

BRYAN BUTTON

CHRIS CLARY

MAX COPE

VERSION HISTORY

Version #	Implemented By	Revision Date	Reason
1.0	Scott Lawson	1/28/2014	Initial Release Converted from Software Plan V1.0

NOTE TO READER

This is a template obtained from:
http://www2.cdc.gov/cdcup/library/templates/default.htm

Template Name: Product Design

UP Template Version: 12/31/07

TABLE OF CONTENTS

1 INTRODUCTION 5
1.1 Purpose of The Document 5
1.2 Overview 5
1.2.1 Objective Statement 5
1.2.2 Theory of Operation 5

2 HARDWARE PLAN 6
2.1 Overview 6
2.2 Level 0 Block Diagram 6
2.3 Level 1 Block Diagram 7
2.4 Level 2 Block Diagram 8
2.5 PSoC3 Block Diagram 9
2.6 Implementation 10
2.6.1 Microcontroller 10
2.6.2 Power Supply 10
2.6.3 Sensors 10

3 SOFTWARE PLAN 11
3.1 Firmware 11
3.1.1 Firmware Overview 11
3.1.2 Initialization Tasks 12
3.1.3 Interrupt Service Routines 13
3.1.4 Low Power ISR 14
3.1.5 USB ISR 14
3.1.6 Data Samples 15
3.2 Computer Interface 15
3.2.1 User Interface 15
3.2.2 Program Installer 15

APPENDIX A: TERMINOLOGY 16
A.1 Acronyms 16
A.2 System Architecture 16

1 INTRODUCTION

1.1 PURPOSE OF THE DOCUMENT

This document describes plans for both the hardware and software components of the Erebus Labs STEM Sensor. It does not include any design files such as schematics, layout files, or source code. Rather, it is intended to guide the team members in their creation of such files and describe how various system components interact with each other.

1.2 OVERVIEW

1.2.1 Objective Statement

Encourage an interest in STEM in K-12 students by delivering a working prototype of an affordable, simple and flexible device to collect environmental data.

1.2.2 Theory of Operation

The Erebus Labs STEM Sensor system is an open-source electronic device for collecting environmental data over a period of time and presenting it for analysis. The system is comprised of the following components:

Base Unit

The central device that manages power, communication, and data storage, and has one or more sensors attached to it.

Sensor

The individual data collection devices such as VOC detectors and thermometers that are attached to the base unit.

User Interface

The program that will be run on a laptop or desktop computer that allows the user to view and interact with the data collected.

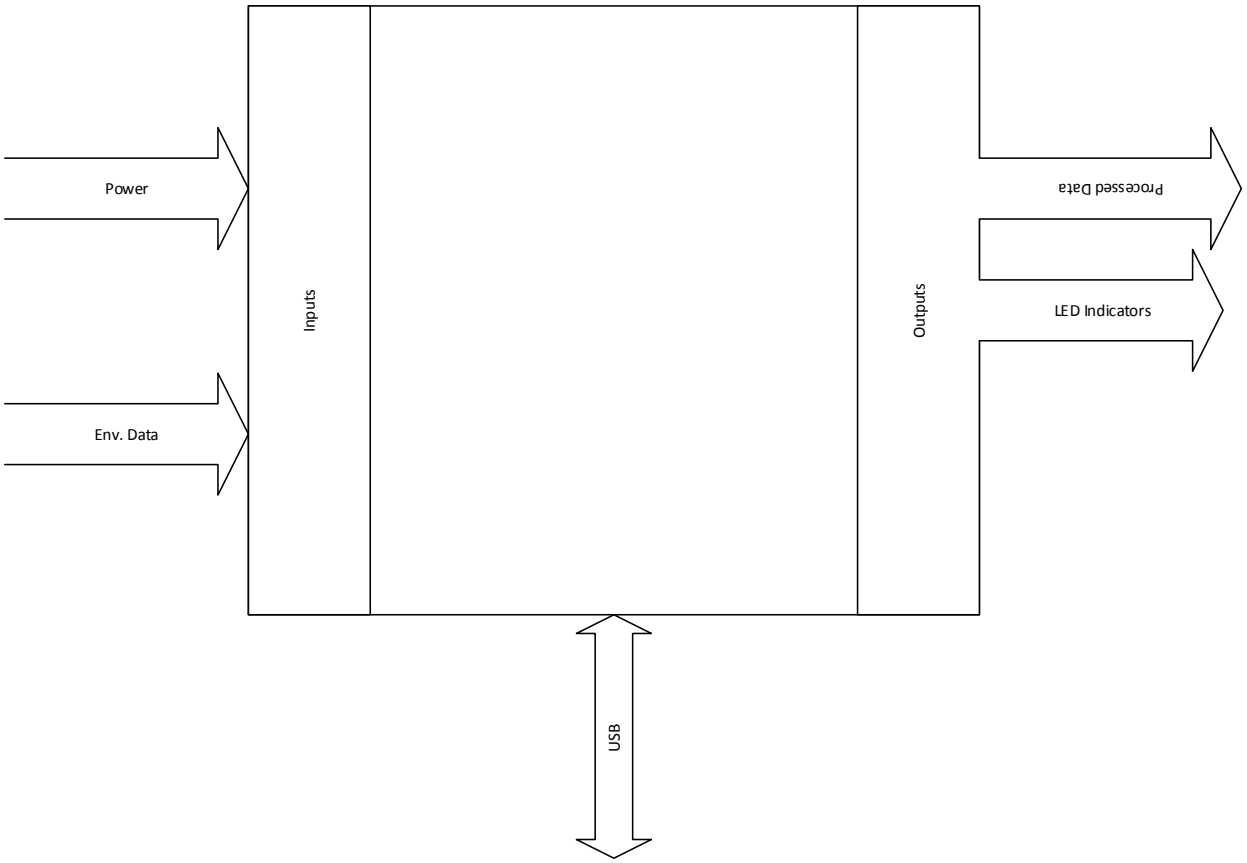
The base unit will have one or more sensors attached to it and will passively collect data without being attached to a computer system. The data collection site will be chosen by the user. The user interface will be a simple GUI for displaying collected data and exporting the data to a CSV file for analysis with a third-party program.

2 HARDWARE PLAN

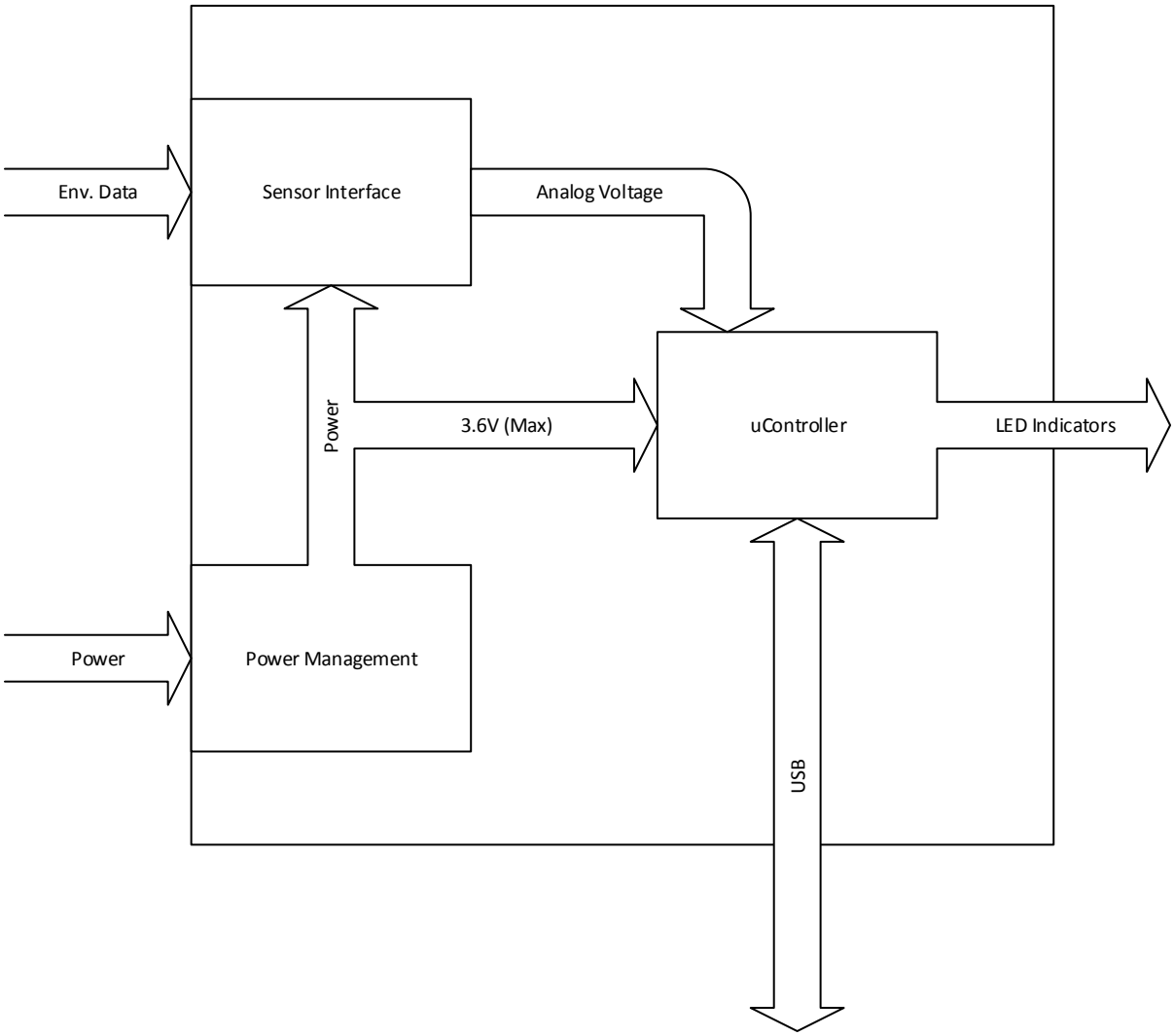
2.1 OVERVIEW

The base unit will be comprised of a Cypress PSoC 3 microcontroller, a voltage regulator, a sensor interface and a USB bus, along with miscellaneous passive and capacitive components.

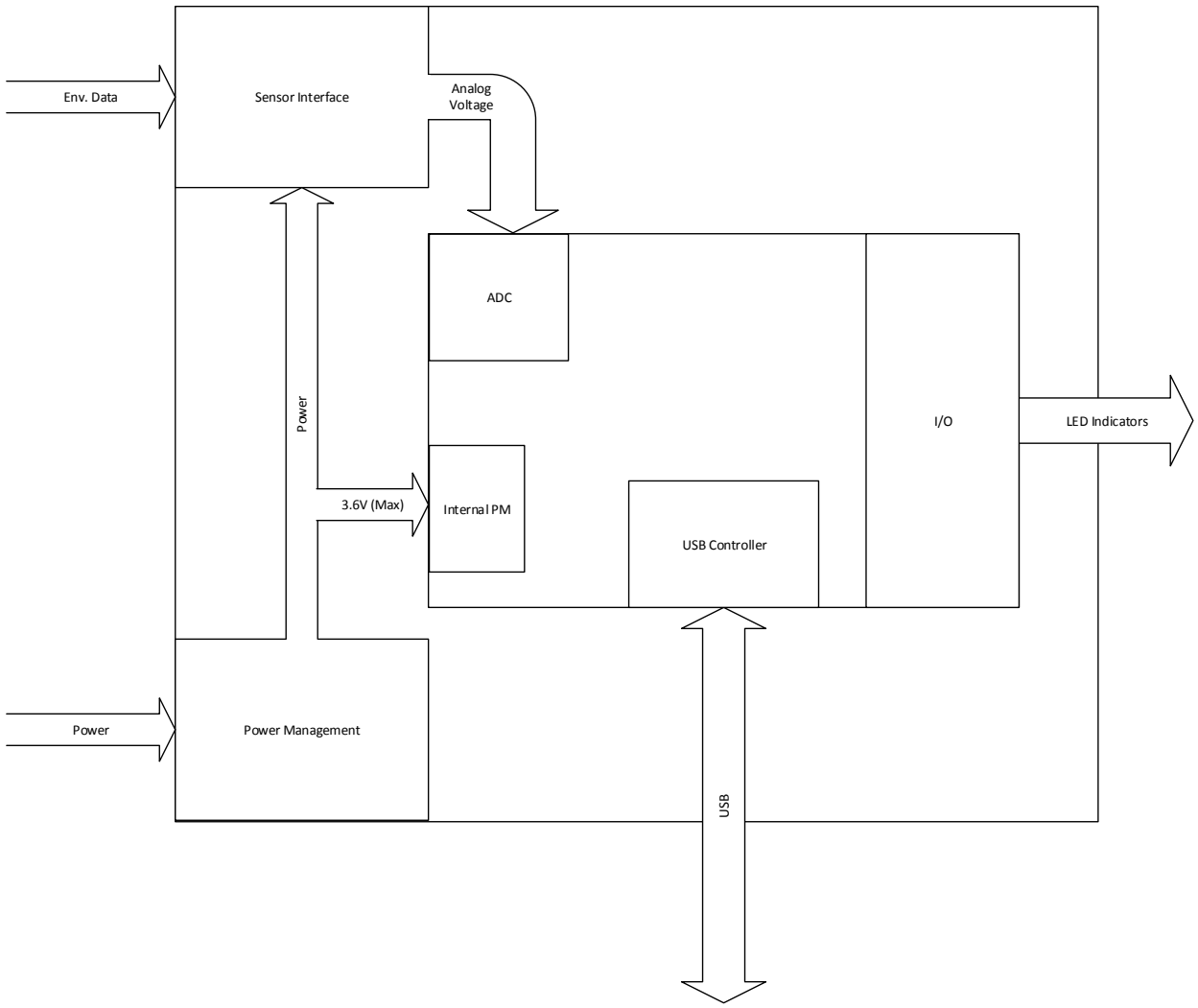
2.2 LEVEL 0 BLOCK DIAGRAM



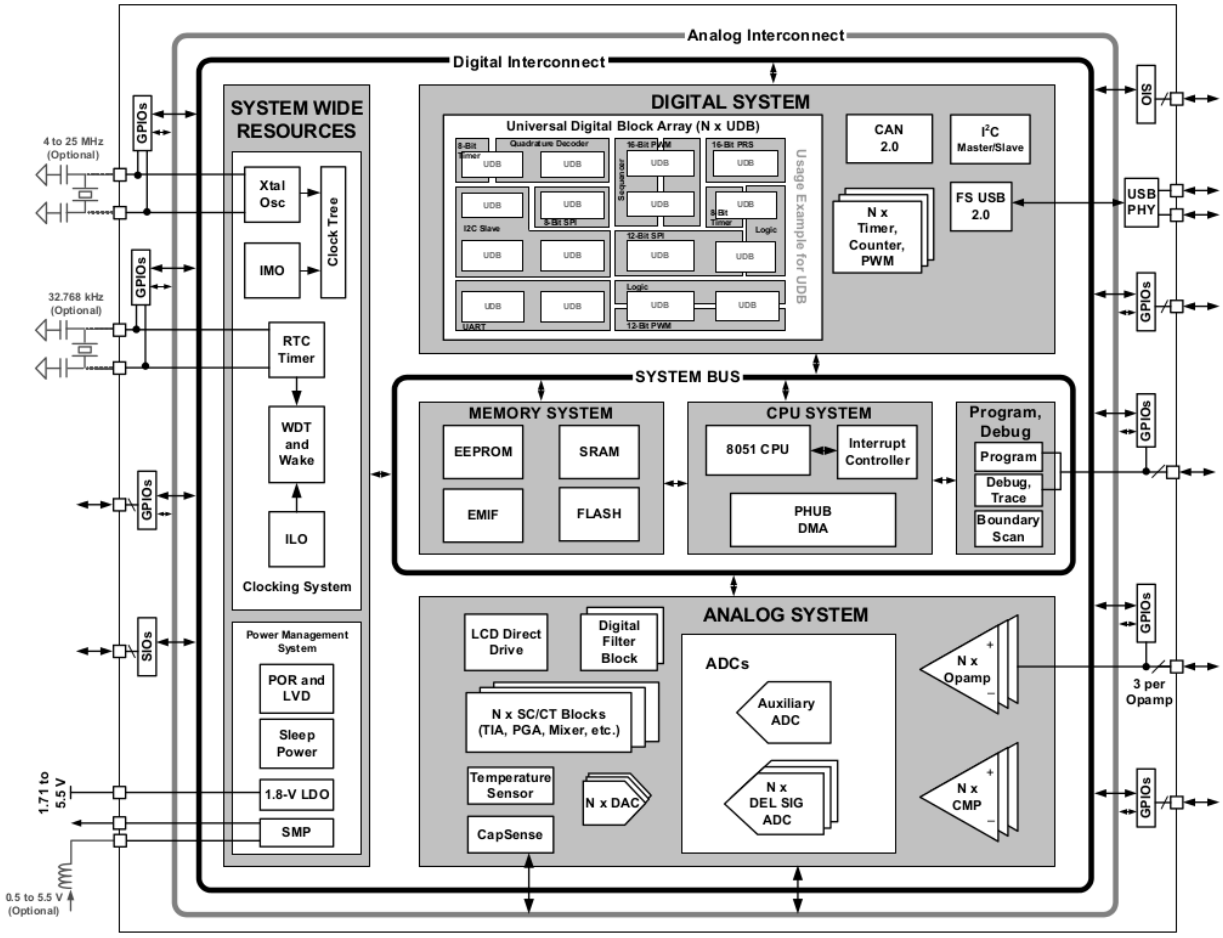
2.3 LEVEL 1 BLOCK DIAGRAM



2.4 LEVEL 2 BLOCK DIAGRAM



2.5 PSOC3 BLOCK DIAGRAM



2.6 IMPLEMENTATION

2.6.1 Microcontroller

The PSoC3 microcontroller from Cypress Semiconductor was selected because of its balance between flexibility and cost. It contains several embedded programmable logic blocks that can be used to implement a Full-Speed USB controller and real-time clock. Additionally, the programmable blocks also provide an interface for utilizing the chip's on-board Flash memory in place of an external EEPROM chip. Therefore, the PSoC3 provides a one-chip solution.

Manufacturer	Cypress Semiconductor
Family	PSoC3
Model Number	CY8C3245PVI-150
Architecture	8-bit 8051
Clock Speed	48MHz
Operating Voltage	1.71V – 5.5V
Current Draw	0.8mA@3MHz, 1.2mA@6MHz, 6.6mA@48MHz
ADC	12-bit Delta-Sigma
Program Memory	32KB Flash
EEPROM	1KB
UDBs	20
Package	48-pin SSOP

2.6.2 Power Supply

No power supply details available at this time.

2.6.3 Sensors

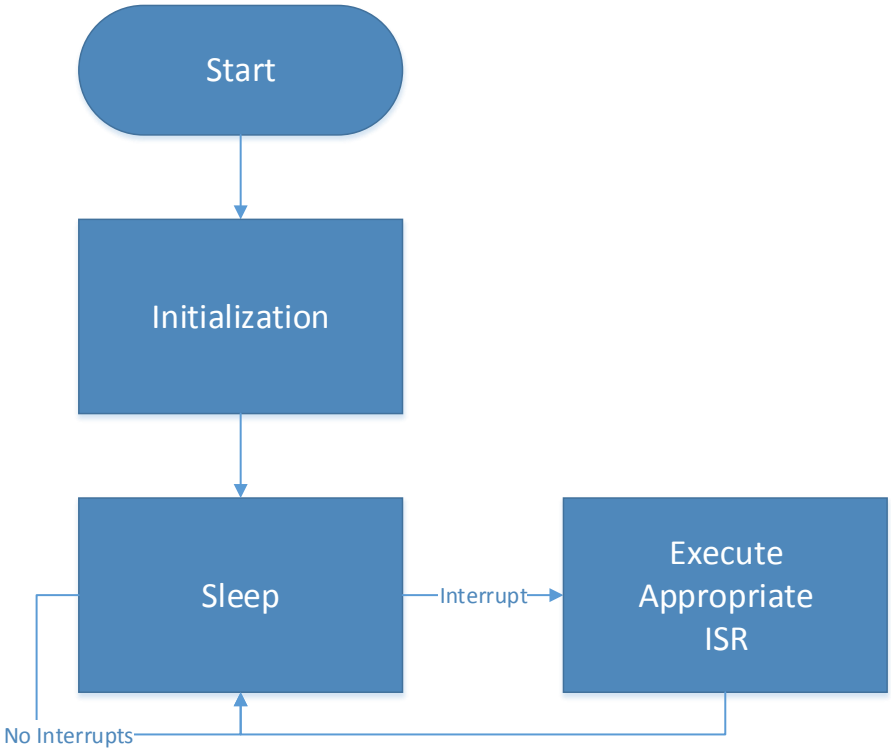
No sensor details available at this time.

3 SOFTWARE PLAN

3.1 FIRMWARE

3.1.1 Firmware Overview

The Erebus Labs STEM Sensors device firmware will be written in standard C compliant with ISO/IEC 9899:2011 and developed in Cypress' PSoC Creator 3.0. Upon reset, the controller will operate as follows:



The STEM Sensor will contain a null-body loop as its MAIN function. All functionality will be interrupt-driven. The system will remain in sleep mode except when connected to a computer via a USB interface, or when sampling the ADC. The system will be woken from sleep by the PSoC3's Central Freewheel Timer to drive ADC sampling.

3.1.2 Initialization Tasks

After a restart, the chip must be initialized. There are several individual components that will require register modifications:

General

- Clocks
- GPIOs including LEDs and buttons

PSoC Components

- USB
    - Set Device / Interface Descriptors
    - Enable USB interrupt assertions from USB controller
- Flash as EEPROM
    - Set wear-leveling pointers
    - Initialize pointers to key data points
- Real Time Clock

Analog to Digital Converter – Delta Sigma

- Enumerate Sensors or retrieve sensor IDs from EEPROM
- Configure clocks, capacitance, gain, and power settings

Central Timewheel

- Retrieve frequency settings from EEPROM
- Branches to ISRs
- Start timer

Power management

- Enable Sleep State
- Power down unused chip components
- Allow waking up from USB (PICU interrupts) and CTW
- Enable voltage monitoring

Interrupts

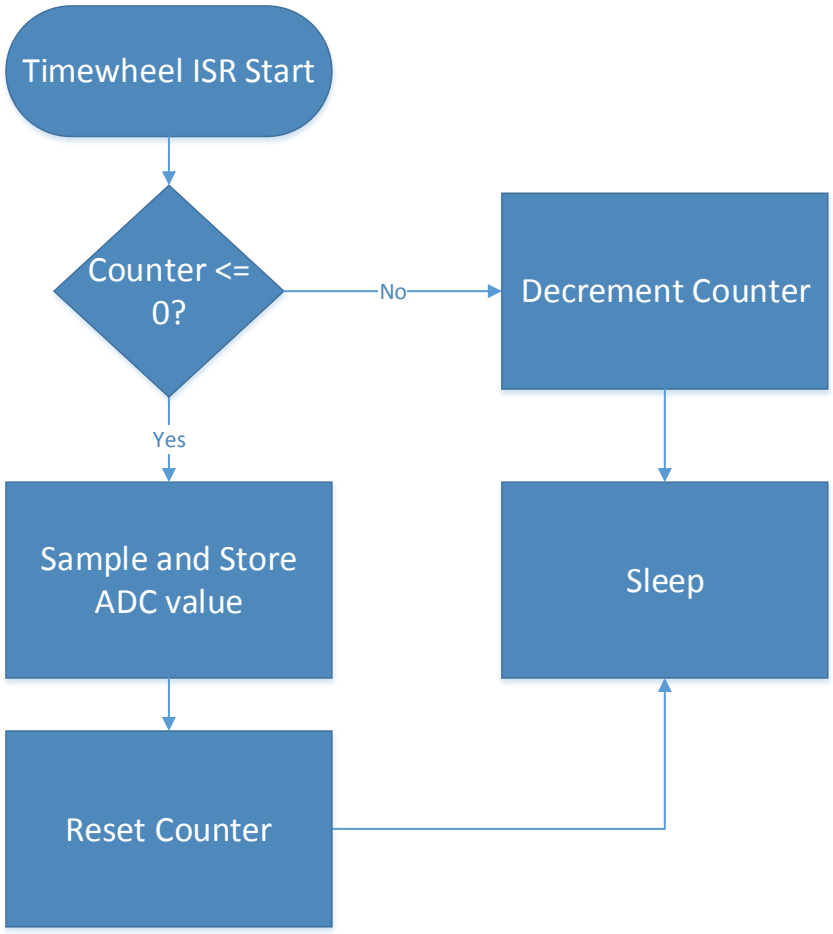
- Set USB (PICU), low voltage, and CTW interrupts in interrupt controller
- Map interrupt vectors to ISRs
- Enable global interrupts

3.1.3 Interrupt Service Routines

Central Timewheel ISR

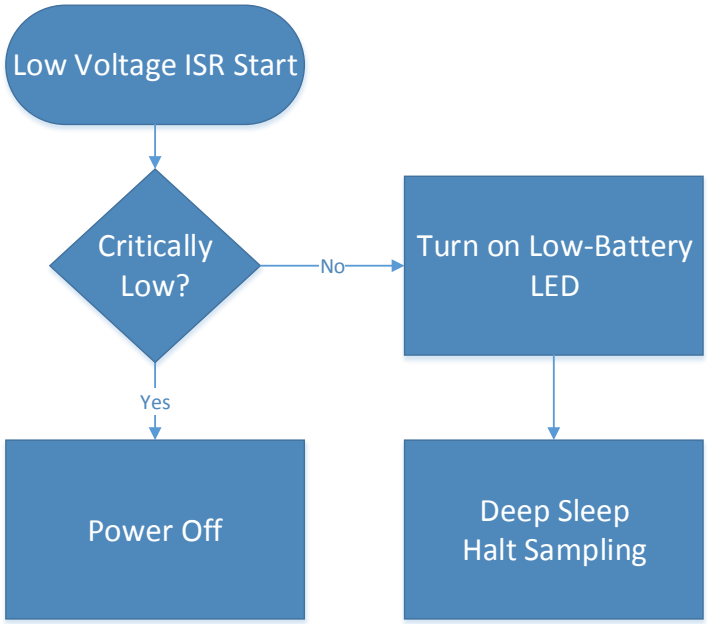
The PSoC3 employs a 1-kHz free-running clock to periodically wake the system up from sleep. The free running clock increments an interval counter that can be set as high as 4096, resulting in the counter rolling over every ~4.096 seconds and generating an interrupt, waking the system from sleep. This value will be adjusted based on the user's desired data sampling frequency.

The firmware will keep a software counter in memory, which will be decremented every time the system is woken from sleep. The value of the counter will be equal to Sample_Period / Wake_Period. See Pg 148 of the PSoC3 TRM for the Central Timewheel information.



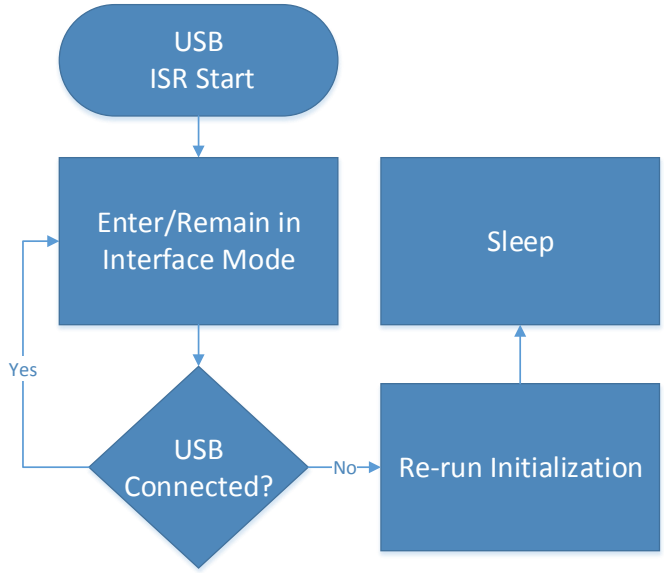
3.1.4 Low Power ISR

The actual power output of the supply batteries cannot be directly measured during operation, so the PSoC3's voltage monitoring capabilities will be utilized to detect low-voltage conditions. The Low Power ISR will determine if the system is critically low on power. If yes, the system will shut down entirely to protect the microcontroller from failure. If no, then turn on a warning LED to the user.



3.1.5 USB ISR

The system will enter an interface mode when plugged into a computer. The sampled data will be transferred to the host computer and interpreted by the user interface software.



3.1.6 Data Samples

To maximize the use of limited memory, the unsigned binary value read directly from the ADC, concatenated with an identifier of the sensor source and a time stamp, and stored raw. The system will not perform conversions or calculations with this data on board the base unit. Every sampled data point will be stored as a 40-bit field.

- DP[39:36] 4-bit sensor identifier
- DP[35:30] Seconds (0-59)
- DP[29:24] Minutes (0-59)
- DP[23:19] Hours (0-23)
- DP[18:14] Days (1-31)
- DP[13:10] Months
- DP[9:0] Data (Least significant 2 bits are discarded)

3.2 COMPUTER INTERFACE

3.2.1 User Interface

The host computer user interface will be a simple GUI that performs three functions:

- Provide a method for the user to change data collection settings or reset the base unit.
- Allow the user export data from the device. When this occurs, the GUI will be responsible for translating the data from the 40-bit fields described in 3.1.6 into human-readable data.
- Allow the user to display simple graphs of previously collected and exported data when the base unit is not connected to the computer.

The interface will be written in Python, compatible with version 3.3 or later.

3.2.2 Program Installer

The user interface software will be packaged in an installer that allows the user's system to recognize the base unit when attached, and launch the GUI.

APPENDIX A: TERMINOLOGY

A.1 ACRONYMS

Acronym	Meaning
ADC	Analog-to-Digital Converter
BOM	Bill of Materials
CO	Carbon Monoxide
CSV	Comma-separated-value formatted file
EEPROM	Electrically Erasable Programmable Read-Only Memory
EPL	The Portland State University Engineering and Prototyping Lab
I^2C	The Inter-Integrated Circuit communication protocol
ISR	Interrupt Service Routine
K-12	Kindergarten through 12^{th} grade school
LED	Light Emanating Diode
PCB	Printed Circuit Board
PSoC	Programmable System On Chip
SI	Silicon
SPI	Serial Peripheral Interface Bus
STEM	Science, Technology, Engineering and Math
TRM	Technical Reference Manual
USB	Universal Serial Bus

A.2 SYSTEM ARCHITECTURE

Base Unit

The central device that manages power, communication, and data storage, and has one or more sensors attached to it.

Sensor

The individual data collection devices such as VOC detectors and thermometers that are attached to the base unit.

User Interface

The program that will be run on a laptop or desktop computer that allows the user to view and interact with the data collected.

System

The operational product comprised of base units with attached sensors and a user interface.