VS1003 AppNote: Using VS1003 Line Input to Monitor Battery Voltage

Description

This document describes how to monitor system battery voltage by using VS1003 Line Input. This document gives general info on the application, shows an example schematic and the necessary register settings.

This document applies to VS1003.

Revision History

Rev	Date	Author	Description
1.00	2006-04-11	PLe	Initial revision.

Table of Contents

1 General

The Line Input of VS1003 can be used to monitor system battery voltage. The ADC in VS1003 is optimized for audio signals but it can also convert DC levels. The input is internally biased near 1.24V. The digital zero value is at this bias point.

The readout of the measurement is inverted around the bias voltage with the maximum readout (32767, 0x7FFF) at 0 V and minimum (-32768, 0x8000) at RCap voltage x 2 (around 2.4 volts). The voltage at Line Input must not exceed AVDD. Input impedance of the Line Input is near 2MOhm at normal clock speeds.

When battery voltage is fed to the Line Input throught appropriate voltege divider it can be easily monitored by seeing if the ADC value is positive or negative.

ADC values can be read from VS1003 through the SCI interface.

Notice that ACPCM recording is not possible while monitoring Line Input voltage.

2 Schematic

A simple voltage divider can be used to provide reference voltage to the Line Input. Assuming the system battery voltage is 3.6V and the low limit for the voltege is 2.4V then the voltage divider should be as shown in figure 1.

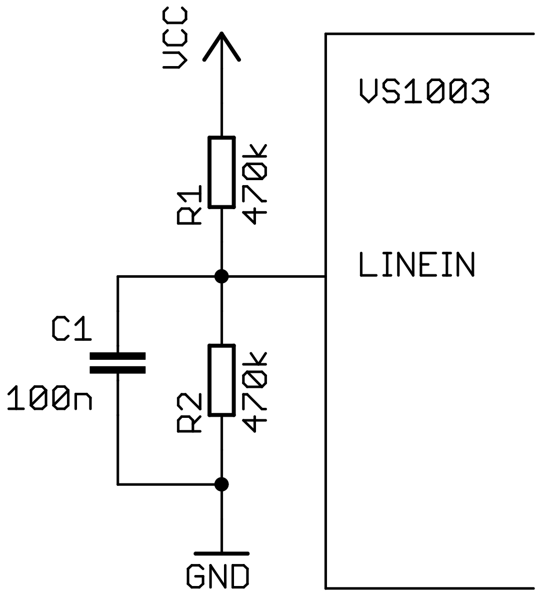

Figure 1: Recommended schematic

R1 should be adjusted so that when VCC drops to the "battery low" limit voltage at Line Input pin is just below 1.2V. As mentioned before the digital zero of the ADC is at about 1.24V.

The 100nF capasitor is used to filter out noise from the input.

3 Reading ADC data

Before values can be read the ADC must be set up properly. This is done by enabling the line input by setting SCI_MODE bit SM_LINE_IN to one. Then by writing a large value to the AD_DIV (0xC01E) register. This sets the ADC to slow conversion speed that saves power. After this, ADC values can be read from AD_DATA (0xC01F) register.

It takes (128 x AD_DIV) clock cycles to gather one sample.

The following pseudo code shows how to set up the ACD and read data through SCI using SCI_WRAMADDR and SCI_WRAM to acces ADC data.

Set up:

```
/* Select Line Input*/
WriteRegister(SCI_MODE, (ReadRegister(SCI_MODE) | SM_LINE_IN)) ;

/* Set AD_DIV to 8192 (23 samples per sec at 24.576MHz internal clock) */
WriteRegister(SCI_WRAMADDR, 0xC01E); /* register to write */
WriteRegister(SCI_WRAM, 0x2000); /* data to write */
```

Data read and compare:

```
/* Read Data From ADC */
WriteRegister(SCI_WRAMADDR, 0xC01F);
adcValue = ReadRegister(SCI_WRAM);

if (adcValue > 100) /* allow offset of 100 */
  BatteryLow();
```

4 Document Version Changes

This chapter describes the most important changes to this document.

Version 1.00, 2006-04-11

- Initial version.

5 Contact Information

VLSI Solution Oy
Hermiankatu 6-8 C
FIN-33720 Tampere
FINLAND

Fax: +358-3-316 5220
Phone: +358-3-316 5230
Email: mp3@vlsi.fi
URL: http://www.vlsi.fi/

Note: If you have questions, first see
http://www.vlsi.fi/vs1001/faq/
http://www.vlsi.fi/vs1011/faq/
http://www.vlsi.fi/vs1002/faq/
http://www.vlsi.fi/vs1003/faq/