

## AN-217 Measuring the Power Supply Voltage

---

### Introduction

Some applications require a means to measure the voltage of the supply that is powering the ZX device. This may be particularly important, for example, in a battery-powered application. At first blush, one might think that the built-in function GetADC() might be useful. This, however, won't work for a reason that is probably not immediately obvious.

The built-in function GetADC() returns a value that is approximately proportional to the voltage on the specified pin but that measurement is made relative to the voltage being applied to the AVcc pin. If the power supply voltage were directly connected to the ADC input pin (no divider) then GetADC() would always return the maximum value no matter what the supply voltage is because the voltage being measured and the reference voltage will always be the same.

Fortunately, the ATmega CPU on which the ZX devices are based provides an internal reference voltage that is relatively constant that can be used as the reference voltage. Using this internal reference voltage is the key to accomplishing the goal of measuring the power supply voltage.

### Using the Internal 2.56V Reference

In order to perform A/D conversions using the internal reference voltage, the internal A/D control registers must be manipulated directly. The function below shows how this can be done.

```
,  
' readADC  
,  
' This function performs an A/D conversion on a specified channel (0-7  
' corresponding to A.0 through A.7) using the ATmega's internal 2.56V  
' reference instead of AVcc. The return value is a Single value that  
' approximates the voltage present on the specified pin. Voltages  
' higher than 2.56V will be read as 2.56V so an external divider  
' circuit may need to be used. For best results, the largest ratio  
' should be used that produces a voltage below 2.56V.  
,  
' The ATmega datasheet should be consulted to better understand the  
' meanings of the registers and bits involved.  
,
```

```
Function readADC(ByVal chan as Byte) as Single
```

```
    ' Wait for the ADC to be available to use.  
    Do While CBool(Register.ADCSRA And &H80)  
        Sleep(1)  
    Loop  
    Call LockTask()  
  
    ' Select the desired channel and the internal 2.56V reference.  
    Register.ADMUX = (chan And &H07) Or &Hc0  
  
    ' Start a conversion using the 128 prescaler (the resulting  
    ' conversion clock must be between 50KHz and 200KHz).  
    Register.ADCSRA = &Hd7
```

## AN-217 Measuring the Power Supply Voltage

```
' Wait for conversion completion. Using the 128 prescaler,  
' the conversion should only take about 0.2mS.  
Do While Not CBool(Register.ADCSRA And &H10)  
    Sleep(1)  
Loop  
  
' Read the conversion result and turn off the converter.  
Dim adcVal as UnsignedInteger  
adcVal = Register.ADC  
Register.ADCSRA = 0  
Call UnlockTask()  
  
' Convert the ADC value to a voltage (2.56V is full scale  
' voltage, 1024 is the number of quantization steps).  
readADC = CSng(adcVal) * (2.56 / 1024.0)  
End Function
```

It is important to note that the internal 2.56V reference may not be exactly 2.56 volts. Variations of the actual reference voltage will affect the result.

## Providing a Voltage to Measure

Since the maximum value that can be returned by the A/D converter when using the 2.56V internal reference represents a value less than 2.56 volts, the power supply voltage will need to be scaled before being applied to the A/D input. This is easily done using a voltage divider as shown in the Figure 1. If the maximum power supply voltage is 5.5 volts, this will yield a maximum voltage to be measured of 2.53 volts.

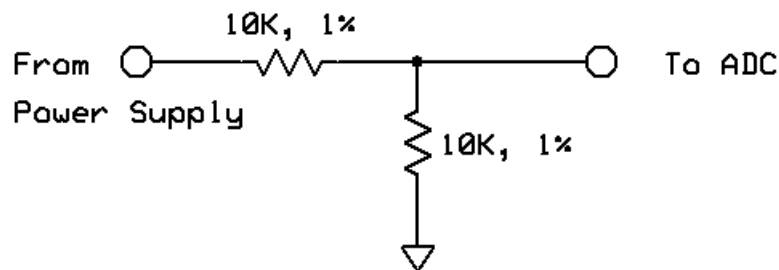


Figure 1 – Voltage Divider

Note that the variations due to use of 1% tolerance resistors in the divider may be too large to meet your needs. In that case, you may need to employ an alternate strategy. One alternative is to employ precision resistors that have a smaller tolerance or to use selected resistors that are closely matched. Another alternative is to use a potentiometer, as illustrated in Figure 2, that is adjusted to the desired ratio during manufacturing test. Of course, this calibration step adds cost to your manufacturing process.

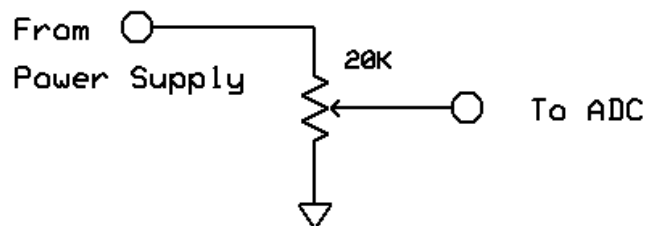


Figure 2 – Potentiometer Divider

## AN-217 Measuring the Power Supply Voltage

With a divider in place, the value returned by readADC() must be properly scaled to reflect the actual supply voltage. The code below illustrates how this is done and how the conversion routine is invoked.

```
Sub Main()  
  Const PS_Bit as Byte = 7          ' to use A.7  
  Do  
    ' scale the result assuming a 2:1 voltage divider  
    Debug.Print "Voltage = "; Fmt(readADC(PS_Bit) * 2.0, 3)  
    Call Delay(1.0)  
  Loop  
End Sub
```

## Summary

This application note illustrates a way to use one of the special features of the on-board analog/digital converter. The A/D converter has other features, such as differential conversions, that may be useful in some applications. The techniques used in this particular application can be modified to exploit those additional features. See the applicable ATmega datasheet for more detailed information.

## Author

Don Kinzer is the founder and CEO of Elba Corporation. He has many years experience working with microprocessors, microcontrollers and general purpose computers. Don can be contacted via email at [dkinzer@zbasic.net](mailto:dkinzer@zbasic.net)

e-mail: [support@zbasic.net](mailto:support@zbasic.net)

Web Site: <http://www.zbasic.net>

**Disclaimer:** Elba Corp. makes no warranty regarding the accuracy of the information in this document nor any warranty in regard to fitness for any particular purpose of the information presented and the techniques described. Furthermore, no warranty is made for the use of the Company's products, other than those expressly contained in the Company's standard warranty which is detailed in the Terms and Conditions statement located on the Company's web site. The Company reserves the right to change the devices, information or specifications described herein at any time without notice, and does not make any commitment to update the information contained herein. No license to any patent or other intellectual property of Elba Corp. is granted by the Company in connection with the sale or use of the Company's products, expressly or by implication. The Company's products are not authorized for use as critical components in life support devices or systems or any other system in which failure or errant operation may endanger life or cause bodily injury.

Copyright © 2005 Elba Corporation. All rights reserved. ZBasic, ZX-24, ZX-24a, ZX-40, ZX-40a, ZX-44, ZX-44a and combinations thereof are trademarks of Elba Corp. or its subsidiaries. Other terms and product names may be the trademarks of others.