Micromint, Inc	**AN752**
	Micro64A/128A
Optional 2-Channel, 12-bit ADC	12/3/04

Introduction: This application note demonstrates how to read the optional 12-bit ADC for the Micro64A/128A.

Background: Micro64A/128A has a built in 2-channel 12-bit ADC. ADC0 (pin 30) and ADC1 (pin 31) is where the analog signals should be connected.

How it works: There are two ways the 12-bit ADC can be read, single ended and differential. The +5V (pin 40) is the reference for the 12-bit ADC. Micro64A/128A has four utility calls for the 12-bit ADC. They are single ended ADC0, single ended ADC1, differential ADC0 subtracted by ADC1, and differential ADC1 subtracted by ADC0. Single ended ADC0 will send back the digital count for the voltage on ADC0. Single ended ADC1 will send back the digital count for the voltage on ADC1. Differential ADC0 subtracted by ADC1 will send back the digital count for ADC0-ADC1. Differential ADC1 subtracted by ADC0 will send back the digital count for ADC1-ADC0. When the differential is read and the result should be negative then a digital count of zero will be sent back. This is because the 12-bit ADC does not have a sign bit and can only send positive numbers. The BASCOM-AVR program listed below is for the Micro64A. Please download the zip file that contains the program for the Micro128A.

Program Listing:

```
'********************************************************************
'Project : Demo program on how access Micro64's MCP3202 ADC.
'Company : Micromint, Inc.
'
'********************************************************************

$regfile = "m64def.dat"
$baud1 = 9600


'Configure the serial port.
Config Com2 = Dummy , Synchrone = 0 , Parity = None , Stopbits = 1 , Databits = 8 , Clockpol = 0
'Configure PORTD.6, PORTD.5, PORTD.1, and PORTD.0 as outputs and the rest of the port as inputs.
Ddrd = &B01100011


'Open the serial port
Open "com2:" For Random As #1
Dim Dat As Word At &HFFE
Dim Adc0 As Word
Dim Adc1 As Word
Dim Diff0 As Word
Dim Diff1 As Word
Dim Vadc0 As Single
Dim Vadc1 As Single
Dim Vdiff0 As Single
Dim Vdiff1 As Single

Dim Vref As Single


  Portd.6 = 1                              'Enable the transmitter
  Waitms 10                                'Wait for the transmitter to settle.
  Print #1 , "Please enter the reference Voltage and press enter?"
  Print #1 ,
```

```
   Portd.6 = 0                          ' Disable the transmitter
   Input #1 , Vref
Do
   $asm
     !Call $7C04
   $end Asm
   Adc0 = Dat
  $asm
     !Call $7BFD
   $end Asm
   Adc1 = Dat
   $asm
     !Call $7C0B
   $end Asm
   Diff0 = Dat
   $asm
     !Call $7C12
   $end Asm
   Diff1 = Dat
   Vadc0 = Vref / 4096
   Vadc1 = Vadc0 * Adc1
   Vdiff0 = Vadc0 * Diff0
   Vdiff1 = Vadc0 * Diff1
   Vadc0 = Vadc0 * Adc0
   Portd.6 = 1                          'Enable the transmitter
   Waitms 10                             'Wait for the transmitter to settle.
   Print #1 , "ADC0 = " ; Fusing(vadc0 , "#.&&") ; " ADC1 = " ; Fusing(vadc1 , "#.&&") ; " Diff0 = " ; Fusing(vdiff0 , "#.&&") ; " Diff1 = " ;
Fusing(vdiff1 , "#.&&")
   Print #1 ,
   Portd.6 = 0                          ' Disable the transmitter
Loop

Close #1


End
```