Micromint, Inc	**AN753**
	Micro64/128
32k SRAM	2/14/05

Introduction: Micro64/128 has a total of 36k of SRAM. 4 k of SRAM is built into the processor an additional 32k of SRAM is available inside the Micro64/128 module but external to the processor. For discussion purposes we will call the 4k that resides in the processor as internal SRAM and the additional 32k of SRAM as external SRAM. The internal SRAM's address runs from 0000 Hex to 10FF Hex. The external SRAM runs from address 1100 Hex to 7FFF Hex. Figure 1 shows a picture of the SRAM memory map. Looking at this memory map leads you to belive that there is only a total of 32k of SRAM. Looks can be deceiving. There is 4k of external SRAM hidden under the internal SRAM. This application note demonstrates how to access all of the external memory.

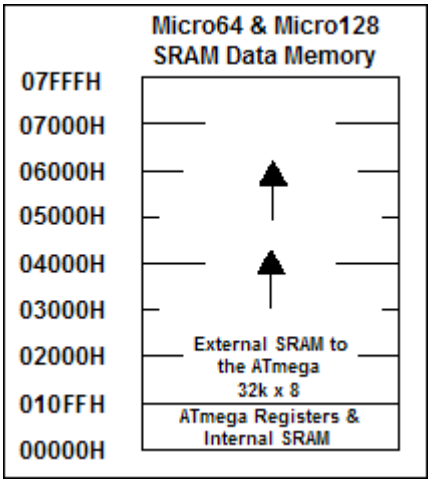

Figure 1: Micro64/128 SRAM Memory Map

Background: : Micro64/128 has a total of 32k of SRAM. Even though there is 32k SRAM external to the mega64 or mega128 the 4 k of internal SRAM masks the lower 4k of the external 32k SRAM. The external SRAM runs from address 1100 Hex to 7FFF Hex.

How it works: In order to access the additional 32k of SRAM the processor needs to be initialized properly so it knows that there is 32k of SRAM external to it. The BASCOM AVR compiler needs to be set up so properly or it will give you errors when the program is compiled. Please follow the following steps to set up the compiler properly:

1. Click on "Options".
2. Highlight "Compiler".
3. Click on "Chip" The following window should appear:

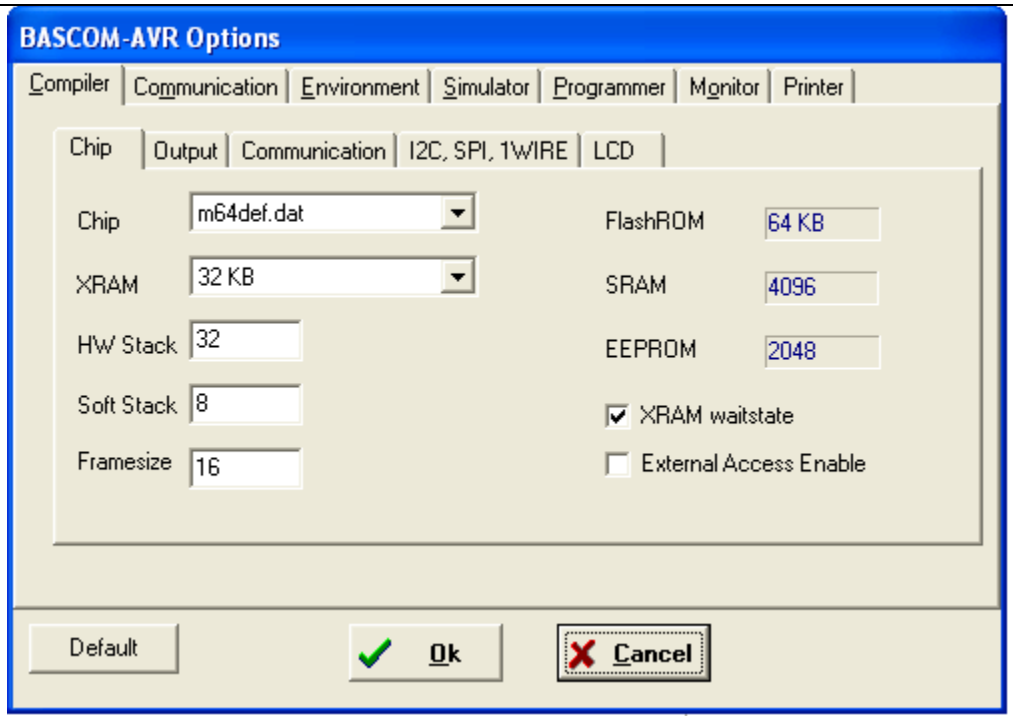

Figure 2: Micro64 BASCOM AVR Set up

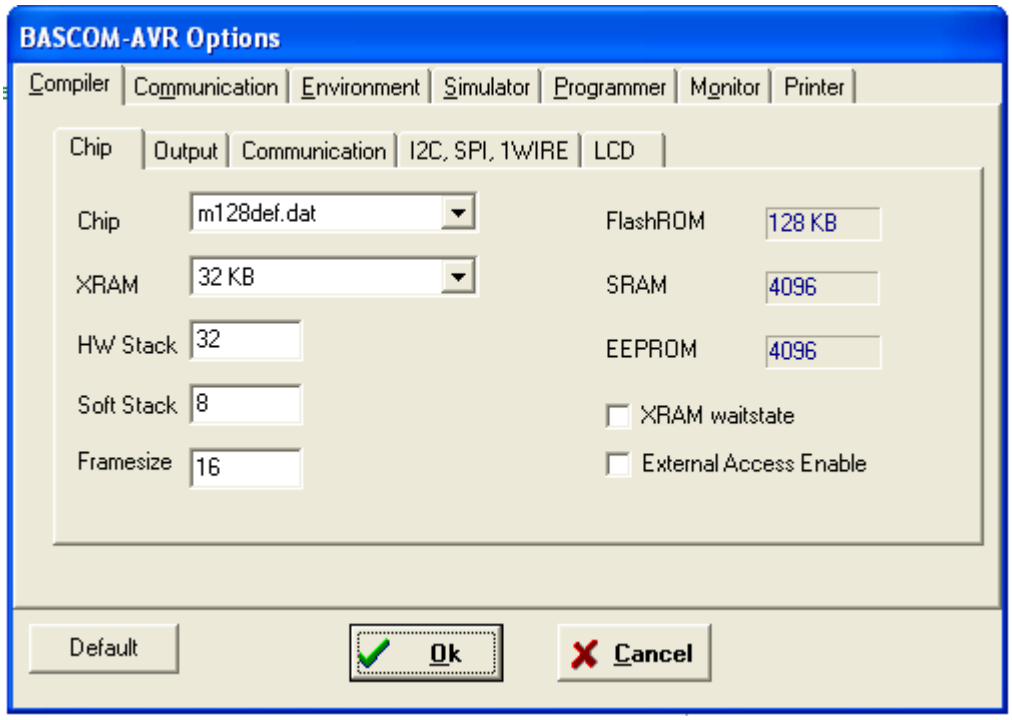

Figure 3: Micro128 BASCOMM AVR Setup

Program Listing:

```
'********************************************************************
'Project : Demo program to show how to use the 32k SRAM in Micro64.
'Company : Micromint, Inc.
'
'********************************************************************

$regfile = "m64def.dat"
$baud1 = 9600
$xramstart = &H1100
$xramsize = &H7FFF
```

```
'Configure the serial port.
Config Com2 = Dummy , Synchrone = 0 , Parity = None , Stopbits = 1 , Databits = 8 , Clockpol = 0
'Configure PORTD.6 as an ouput and the rest of the port as inputs.
Ddrd = 64


'Open the serial port
Open "com2:" For Random As #1
Dim T As Byte
Dim W As Byte
Dim X As Xram String * 50 At &H2001
Dim K As Xram Byte At &H2000
Declare Sub WRRAM_0000h(BYVAL MemAddr As Word , WrData As Byte)
Declare Sub RDRAM_0000h(BYVAL MemAddr As Word)
K = 100
Portd.6 = 1                 'Enable the transmitter
  Waitms 10                 'Wait for the transmitter to settle.
   Print #1 , "K = " ; K
   Print #1 ,
Do
   K = 230
   Print #1 , "K = " ; K
   Print #1 , "Please enter a Byte!"
   Print #1 ,
   Input #1 , K
   Print #1 , "Please enter a your name!"
   Print #1 ,
   Input #1 , X
   CALL RDRAM_0000h(&H0001)
   CALL RDRAM_0000h(&H0002)
   CALL RDRAM_0000h(&H0003)
   W = 11
   Call WRRAM_0000h(&H0001 , W)
   W = 22
   Call WRRAM_0000h(&H0002 , W)
   W = 33
   Call WRRAM_0000h(&H0003 , W)
   CALL RDRAM_0000h(&H0001)
   CALL RDRAM_0000h(&H0002)
   CALL RDRAM_0000h(&H0003)
   Print #1 , X;
   Print #1 , " entered " ; K;
   Print #1 , " for the variable K."
   Print #1 ,

Loop
End

Sub WRRAM_0000h(MemAddr As Word , WrData As Byte)
DDRC = &HFF
PORTC = 0
XMCRB.0 = 1
XMCRB.1 = 1
OUT MemAddr , WrData
XMCRB.0 = 0
XMCRB.1 = 0
PRINT #1 , "Wrote " ; WrData ; " from address " ; MemAddr
End Sub

Sub RDRAM_0000h(MemAddr As Word)
DDRC = &HFF
PORTC = 0
XMCRB.0 = 1
XMCRB.1 = 1
T = INP(MemAddr)
XMCRB.0 = 0
XMCRB.1 = 0
PRINT #1 , "Read " ; T ; " from address " ; MemAddr

End SUB

Close #1
```