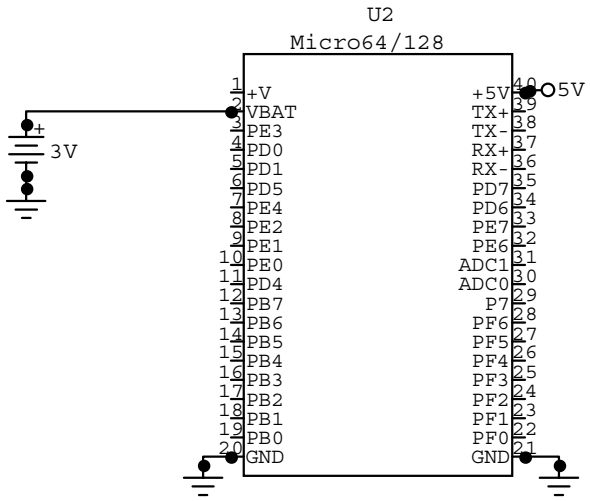
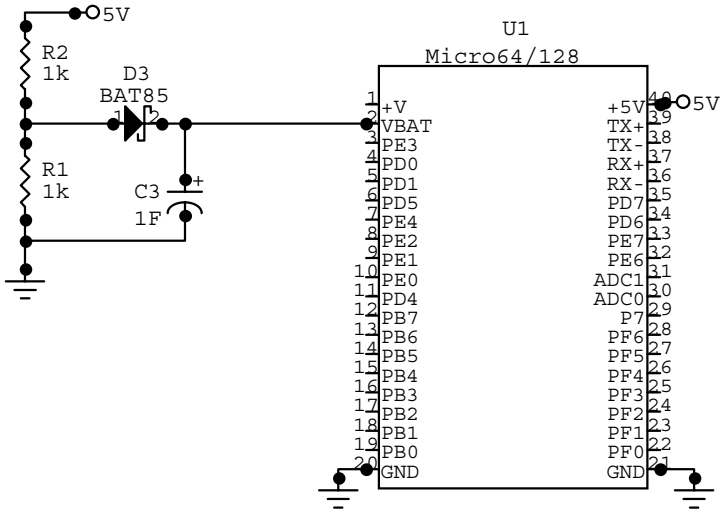
Micromint, Inc	**AN754**
	Micro64/128
I²C Real Time Clock Calendar	12/3/04

Introduction: This application note demonstrates how to read and write the to the I²C Real Time Clock Calendar. It also demonstrates how to use a Coin Battery or a Super Capacitor for battery backup.

Background: Micro64/128 has an I²C Real Time Clock (RTC) built into it. There are two ways that the clock can be backed up so that it continues to increment if power is lost, by battery or by super capacitor. The battery is pretty straight forward. All you have to do is connect a 3V battery to VBAT. The following schematic demonstrates connecting a battery to the Micro64/128.



The following schematic demonstrates connecting a super capacitor to VBAT:



How it works: The RTC uses the I^2C bus to communicate to the mega64 or mega128 controller. The Micro64/128 has utilities for the following functions for the RTC:

Read & Write Functions	Write Only Functions
Tenth of a Second Register	Enable The Alarm
Seconds Register	Disable the Alarm
Minutes Register	Set Alarm to Repeat every Second
Hours Register	Set Alarm to Repeat every Minute
Day of the Week Register	Set Alarm to Repeat every Hour
Day of the Month Register	Set Alarm to Repeat every Day
Month	Set Alarm to Repeat every Month
Year Register	Set Alarm to Repeat every Year
Alarm Seconds Register	**Read Only Functions**
Alarm Minutes Register	Read the Alarm Flags
Alarm Hours Register	
Alarm Day of the Month Register	

The RTC's interrupt pin is connected to PORTE bit 5(PE5). The BASCOM AVR program demonstrates how to use the Micro64/128's utilities to access the RTC.

Program Listing:

```
'********************************************************************
'Project : Demo program on how to set and read Micro64's Real Time Clock
'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 as an ouput and the rest of the port as inputs.
Ddrd = 64
On Int5 Int5_int                        'Initialise the INT5 Interrupt

Enable Interrupts
Disable Int5                            'disable the interrupt

'Open the serial port
Open "com2:" For Random As #1
Dim Result As Word At &HFFE
Dim Dat As Byte At &HFFD
Dim Tenthsec As Byte
Dim Second As Byte
Dim Minute As Byte
Dim Hour As Byte
Dim Dow As Byte
Dim Dom As Byte
Dim Month As Byte
Dim Year As Byte
Dim B As Byte

Portd.6 = 1                             'Enable the transmitter
Waitms 10                               'Wait for the transmitter to settle.
Do
  Print #1 , "RTC Menu"
  Print #1 , "1 - Set the Time"
  Print #1 , "2 - Read the Time"
  Print #1 , "3 - Set the Alarm"
  Print #1 , "4 - Read the Alarm"
  Print #1 , "5 - Enable the Alarm"
  Input #1 , B
  Select Case B
   Case 1 : Gosub Setclk
```

```
    Case 2 : Gosub Readclock
    Case 3 : Gosub Setalarm
    Case 4 : Gosub Readalarm
    Case 5 : Gosub Enablealarm
  End Select

Loop



Setclk:
   Print #1 , "Please enter the Seconds."
   Input #1 , Second
   Second = Makebcd(second)
   Print #1 , "Please enter the Minutes."
   Input #1 , Minute
   Minute = Makebcd(minute)
   Print #1 , "Please enter the Hours."
   Input #1 , Hour
   Hour = Makebcd(hour)
   Print #1 , "Please enter the Day of the Week. 1 = Sunday."
   Input #1 , Dow
   Dow = Makebcd(dow)
   Print #1 , "Please enter the Month."
   Input #1 , Month
   Month = Makebcd(month)
   Print #1 , "Please enter the Day of the Month."
   Input #1 , Dom
   Dom = Makebcd(dom)
   Print #1 , "Please enter the Year."
   Input #1 , Year
   Year = Makebcd(year)
   Dat = Second
   $asm
     !Call $7D15
   $end Asm
   Dat = Minute
   $asm
     !Call $7D1D
   $end Asm
   Dat = Hour
   $asm
     !Call $7D21
   $end Asm
   Dat = Dow
   $asm
     !Call $7D29
   $end Asm
   Dat = Month
   $asm
     !Call $7D2D
   $end Asm
   Dat = Dom
   $asm
     !Call $7D31
   $end Asm
   Dat = Year
   $asm
     !Call $7D35
   $end Asm
Return

Readclock:
Do
   $asm
     !Call $7CF3
   $end Asm
   Tenthsec = Result
   $asm
     !Call $7CF7
   $end Asm
   Second = Result
   $asm
```

```
    !Call $7CFC
  $end Asm
  Minute = Result
  $asm
    !Call $7D00
  $end Asm
  Hour = Result
  $asm
    !Call $7D05
  $end Asm
  Dow = Result
  $asm
    !Call $7D09
  $end Asm
  Dom = Result
  $asm
    !Call $7D0D
  $end Asm
  Month = Result
  $asm
    !Call $7D11
  $end Asm
  Year = Result
  Select Case Dow
   Case 1 : Print #1 , "Sunday ";
   Case 2 : Print #1 , "Monday ";
   Case 3 : Print #1 , "Tuesday ";
   Case 4 : Print #1 , "Wednesday ";
   Case 5 : Print #1 , "Thursday ";
   Case 6 : Print #1 , "Friday ";
   Case 7 : Print #1 , "Saturday ";
 End Select
  Print #1 , Bcd(month) ; "/" ; Bcd(dom) ; "/" ; Bcd(year) ; " " ; Bcd(hour) ; ":" ; Bcd(minute) ; ":" ; Bcd(second) ; ":" ; Bcd(tenthsec) ; Chr(13);
  B = Inkey(#1)
Loop Until B <> 64

Return

Setalarm:
Print #1 , "Please enter the Alarm Seconds."
  Input #1 , Second
  Second = Makebcd(second)
  Print #1 , "Please enter the Alarm Minutes."
  Input #1 , Minute
  Minute = Makebcd(minute)
  Print #1 , "Please enter the Alarm Hours."
  Input #1 , Hour
  Hour = Makebcd(hour)
  Print #1 , "Please enter the Alarm Month."
  Input #1 , Month
  Month = Makebcd(month)
  Print #1 , "Please enter the Alarm Day of the Month."
  Input #1 , Dom
  Dom = Makebcd(dom)
  Month.7 = 1

  Dat = Dom
  $asm
    !Call $7D91
  $end Asm
  Dat = Month
  $asm
    !Call $7DA4
  $end Asm
  Dat = Hour
  $asm
    !Call $7D7E
  $end Asm
  Dat = Minute
  $asm
    !Call $7D6B
  $end Asm
  Dat = Second
  $asm
    !Call $7D58
```

```
  $end Asm

Return
Readalarm:
  $asm
    !Call $7D39
  $end Asm
  Second = Result
  $asm
    !Call $7D3E
  $end Asm
  Minute = Result
  $asm
    !Call $7D43
  $end Asm
  Hour = Result
  $asm
    !Call $7D4D
  $end Asm
  Month = Result
  $asm
    !Call $7D48
  $end Asm
  Dom = Result
  Print #1 , Bcd(month) ; "/" ; Bcd(dom) ; " " ; Bcd(hour) ; ":" ; Bcd(minute) ; ":" ; Bcd(second)
Return

Enablealarm:
  Print #1 , "Enable RTC Alarm Menu"
  Print #1 , "1 - Every Second"
  Print #1 , "2 - Every Minute"
  Print #1 , "3 - Every Hour"
  Print #1 , "4 - Every Day"
  Print #1 , "5 - Every Month"
  Print #1 , "6 - Every Year"
  Print #1 , "7 - Disable the Alarm"
  Input #1 , B
  Select Case B
   Case 1 :
         $asm
          !Call $7DD2
         $end Asm
         $asm
          !Call $7DB9
         $end Asm
   Case 2 :
         $asm
          !Call $7DFB
         $end Asm
         $asm
          !Call $7DB9
         $end Asm
   Case 3 :
         $asm
          !Call $7D24
         $end Asm
         $asm
          !Call $7DB9
         $end Asm
   Case 4 :
         $asm
          !Call $7D4D
         $end Asm
         $asm
          !Call $7DB9
         $end Asm
   Case 5 :
         $asm
          !Call $7D76
         $end Asm
         $asm
          !Call $7DB9
         $end Asm
   Case 6 :
         $asm
```

```
          !Call $7DA2
        $end Asm
        $asm
          !Call $7DB9
        $end Asm
    Case 7 :
        $asm
          !Call $7DC7
        $end Asm
  End Select
  Enable Int5
Return
Int5_int:
    Disable Int5
    PUSH R31
    PUSH R30
    PUSH R29
    PUSH R28
    PUSH R27
    PUSH R26
    PUSH R25
    PUSH R24
    PUSH R23
    PUSH R22
    PUSH R21
    PUSH R20
    PUSH R19
    PUSH R18
    PUSH R17
    PUSH R16
    PUSH R15
    PUSH R14
    PUSH R13
    PUSH R12
    PUSH R11
    PUSH R10
    PUSH R9
    PUSH R8
    PUSH R7
    PUSH R6
    PUSH R5
    PUSH R4
    PUSH R3
    PUSH R2
    PUSH R1
    PUSH R0
    Portd.6 = 1                         'Enable the transmitter
    Waitms 10
    Print #1 , "Alarm Alarm"
    $asm
      !Call $FED0
    $end Asm
    POP R0
    POP R1
    POP R2
    POP R3
    POP R4
    POP R5
    POP R6
    POP R7
    POP R8
    POP R9
    POP R10
    POP R11
    POP R12
    POP R13
    POP R14
    POP R15
    POP R16
    POP R17
    POP R18
    POP R19
    POP R20
    POP R21
    POP R22
```

```
    POP R23
    POP R24
    POP R25
    POP R26
    POP R27
    POP R28
    POP R29
    POP R30
    POP R31
    Enable Int5
Return
Close #1

End
```