

```

{ ****
*
*                               R T C P
*
*-----
* Task           : Provides two functions for accessing the
*                 battery operated realtime clock.
*-----
* Author        : Michael Tischer
* Developed on   : 07/10/87
* Last update    : 02/27/92
*-----
* ****
}

```

program RTCP;

```

Uses Crt;                                { Add CRT unit }

const RTCAAdrPort = $70;                  { RTC address register }
      RTCDtaPort  = $71;                  { RTC data register }

      SECONDS     = $00;  { Addresses for some RTC memory locations }
      MINUTE      = $02;
      ANHOUR      = $04;
      DAY         = $07;
      MONTH       = $08;
      YEAR        = $09;
      STATUSA     = $0A;
      STATUSB     = $0B;
      STATUSE     = $0C;
      STATUSD     = $0D;
      DIAGNOSE    = $0E;
      HUNDREDEYE = $32;

{ ****
}

```



```

begin
  if (RTCRead(STATUSB) and 2 = 0)           { BCD or binary mode? }
  then RTCDT := RTCRead(Address)             { Binary }
  else                                           { BCD }
  begin
    SVal := RTCRead(Address);                { Get contents of memory location }
    RTCDT := (SVal shr 4) * 10 + SVal and 15   { Convert to binary }
  end
end;

{*****}
{ * RTCWrite: Writes a value to the RTC memory location. * }
{ * Input   : ADDRESS = Address of memory location in the RTC * }
{ *          CONTENT = New value for this memory location * }
{ * Output  : None * }
{ * Info   : This address should range from 0 to 63 * }
{*****}

procedure RTCWrite(Address : integer;      { Address of memory location }
                   Content  : byte);       { New contents }

begin
  Port[RTCAdrPort] := Address;             { Pass RTC address }
  Port[RTCDtaPort] := Content              { Write new value }
end;

{*****}
{ *                               MAIN PROGRAM                               * }
{*****}

begin
  ClrScr;                                  { Clear screen }

```

```

writeln('RTCP (c) 1987, 92 by Michael Tischer'#13#10);
writeln('Information from the battery operated realtime clock');
writeln('===== '#13#10);
if RTCRead(Diagnose) and 128 = 0 then      { Is the battery O.K.? }
begin                                     { Yes --> Battery O.K. }
    writeln('- The clock is in ', (RTCRead(STATUSB) and 2)*6+12,
        ' hour mode');
    writeln('- The time : ', RTCDT(AN HOUR), ':', RTCDT(MINUTE):2,
        ':', RTCDT(SECONDS):2);
    write('- The date : ');
    writeln(RTCDT(MONTH), '-', RTCDT(DAY), '-',
        RTCDT(HUNDREDEYEAR), RTCDT(YEAR));
end
else                                     { Dead battery }
    write('          Attention! Clock battery is dead')
end.

```