

C listing: RTCC.C

```

/*****
/*
/*          R T C C
/*
/*-----*/
/*  Task          : Provides two functions for accessing the
/*                  battery operated realtime clock.
/*-----*/
/*  Author         : Michael Tischer
/*  Developed on   : 07/10/87
/*  Last update    : 04/07/95
*****/
/*  (MICROSOFT C)
/*  Compilation   : CL /AS RTCC.C
/*  Call          : RTCC
/*-----*/
/*  (BORLAND TURBO C)
/*  Compilation   : Use the RUN command (no project file needed)
*****/

/*== Include files =====*/

#include <dos.h>
#include <conio.h>
#include <stdio.h>

/*== Type declarations =====*/

typedef unsigned char BYTE;

/*== Constants =====*/

#define RTCAdrPort    0x70                /* RTC address register */
```

```

#define RTCDtaPort      0x71          /* RTC data register */

#define SECONDS         0x00          /* Addresses for some */
#define MINUTE          0x02          /* RTC memory locations */
#define ANHOUR          0x04
#define DAY             0x07
#define MONTH           0x08
#define YEAR            0x09
#define STATUSA         0x0A
#define STATUSB         0x0B
#define STATUSC         0x0C
#define STATUSD         0x0D
#define DIAGNOSE        0x0E
#define HUNDREDEYEAR    0x32

/*****
/* RTCRead : Reads one of the RTC memory locations.          */
/* Input   : ADDRESS = Memory location in RTC                */
/* Output  : Contents of the memory location                 */
/* Info    : If the address lies outside the valid range (0-63), */
/*           the value -1 is returned.                        */
*****/

BYTE RTCRead(BYTE ADDRESS)
{
    outp(RTCAdrPort, ADDRESS);          /* Send address in RTC */
    return(inp(RTCDtaPort));            /* Get contents of RTC */
}

/*****
/* RTCDT    : Reads a BCD date or time memory location from the RTC, */
/*           and converts the value to a binary value.                */
*****/

```

```

/* Input   : ADDRESS = Address of memory location in the RTC      */
/* Output  : Contents of this memory location in binary notation  */
/* Info    : If the address lies outside the valid range (0-63),  */
/*           the value -1 is returned.                             */
/*****

```

```

BYTE RTCDt(BYTE ADDRESS)

```

```

{
    if (RTCRead(STATUSB) & 2) /* BCD or binary mode? */
        return((RTCRead(ADDRESS) >> 4) * 10 + (RTCRead(ADDRESS) & 15));
    else return(RTCRead(ADDRESS)); /* Binary */
}

```

```

/*****
/* 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              */
/*****
void RTCWrite(BYTE ADDRESS, BYTE Size)

```

```

{
    outp(RTCAdrPort, ADDRESS); /* Send RTC address */
    outp(RTCDtPort, Size);    /* Write new value */
}

```

```

/*****
/**                                MAIN PROGRAM                                **
/*****

```

```

void main()

{

printf("\nRTCC (c) 1987, 1992 by Michael Tischer\n\n");
printf("Information from the battery operated realtime clock\n");
printf("=====\\n\\n");
if (!(RTCRead(DIAGNOSE) & 128))          /* Is the clock O.K.? */
{                                          /* O.K. */
    printf("- The clock is in %d hour mode\\n",
        (RTCRead(STATUSB) & 2)*6+12);
    printf("- The time: %2d:%02d:%02d\\n",
        RTCDt(AN HOUR), RTCDt(MINUTE), RTCDt(SECONDS));
    printf("- The date: ");
    printf("%02d-%02d-%d\\n", RTCDt(MONTH), RTCDt(DAY),
        RTCDt(HUNDREDEYEAR), RTCDt(YEAR));
}
else printf("          Attention! The clock battery is dead.\\n");
}

```