AppKit:

Using the DS1302 Trickle Charge Timekeeping Chip

This AppKit shows how to use the Dallas Semiconductor DS1302 Trickle Charge Timekeeping Chip with the Parallax BASIC Stamp® II single-board computer. It is feasible to use the DS1302 with the Stamp I, but this code is under development. E-mail us if you are using a Stamp I at stamptech@parallaxinc.com

Description

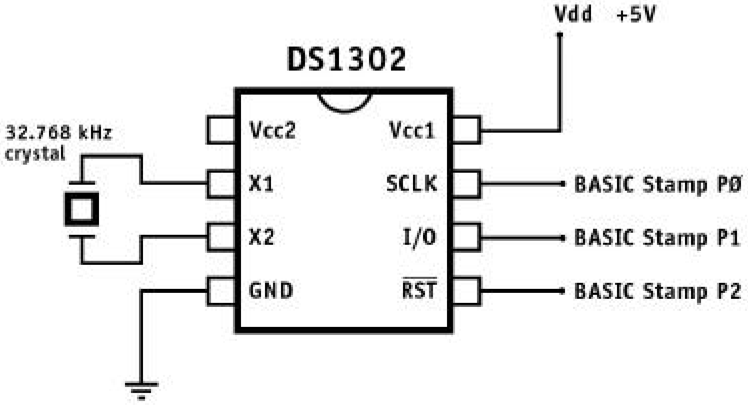
The DS1302 is a real-time clock / calendar with 31 bytes of static RAM. The real time clock counts seconds, minutes, hours, date of the month, month, day of the week, and year with leap year compensation. The DS1302 requires 2.5 – 5.5 volt full operation, and uses less than 300 nA at 2.5 volts. The DS1302 communicates with a microcontroller such as Stamp through a three-wire serial connection.

A temporary connection to a controller establishes the DS1302's time. Thereafter, the chip can operate as a stand-alone clock. This AppKit shows how to program the time into the DS1302, and then allow the clock to operate independently while updating time to the BASIC Stamp. The DS1302 has dual power supply pins for primary and backup, the latter which may be powered by a super cap input. This project relies on the chip's primary power supply input (V_{CC1}).

Hardware interface

The DS1302 interfaces with controllers through a three-wire connection, consisting of a serial clock (SCLK) for data input, input/output line (I/O) for connection to the clock input, and reset (RST) for turning on control logic which accesses the shift register ad provides a method of terminating either single byte or multiple byte data transfer. The power supply pin (V_{CC1}) and ground (GND) may be connected to the Stamps +5V and ground, respectively. The DS1302's X1 and X2 pins are connected to the leads of the 32.768 kHz crystal.[1]

The figure shows how to connect the DS1302 to the Stamp for a demo program which programs the time into the chip, and may then be modified to debug the time to your PC screen.



[1] The DS1302 chip (part #251-03230 priced at $6.00 for quantity one) and 32.768 kHz crystals (part #251-03230 priced at $3.00 quantity one) may also be ordered individually from Parallax .

Software interface

From a software standpoint, using the DS1302 requires only a few steps:

(1) Identify clock starting time using different variable time registers.
(2) Reset the chip and send it an instruction telling it the starting time.
(3) Read the time from the chip and debug it to the PC.
(4) Deactivate RST after each step by taking it low.

The program listings and data sheets show these processes in detail.

Tips for using the DS1302

- Use a capacitor to operate the chip. Place a diode between the DS1302's VCC1 and VCC2 pins. Connect VCC2 to a the 5 volt lead of the capacitor, and connect the negative lead of the cap to ground.. The capacitor will let power through to the DS1302 when power is applied to the board and charge the cap. The cap will discharge it's 5 volts when power is removed and the DS1302 will use the cap as a charge. Depending on the size of the capacitor this charge could last for a few days.

- Use the DS1302's RAM as extra storage space for the Stamp. The 31 bytes could be used for variable storage, and with the cap circuit described above this could be battery backed-up RAM (though it's not non-volatile).

- The DS1302's clock calculates leap years up the year 2100. In order to make this work you must set the day of the week properly to handle the date compensation.

- Solder the crystal leads very close to the DS1302 the chip since any additional lead capacitance will change the timing and make the clock either fast or slow. Always use either of the two crystals recommended by Dallas in the attached data sheet.

BASIC Stamp II (BS2-IC) Program Listing #1

```
'*****************************************************************************
'*      Title: DS1302_1.BS2    Author: Jeff A Martin    Date: 5/18/98        *
'*                                                                           *
'* Description: This BASIC Stamp II program interfaces to the Dallas Semi.            *
'*            DS1302 Real Time Clock (RTC) chip.  The date and time is                *
'*            read and displayed in long and short formats on the debug    *
'*            screen.                                                                  *
'*                                                                           *
'* Notes: This program can be modified to fit into a smaller code space.    *
'*       It is not written as compact as possible to make it more readable   *
'*       and to demonstrate all the useful functions of the chip.                     *
'*       The DS1302 features seconds, minutes, hours (AM/PM-12/24 modes),             *
'*       date of month, month, day of week and year time-keeping with        *
'*       leap year compensation valid up to 2100.  Scratchpad RAM memory              *
'*       (31 bytes), single-byte and multi-byte reads and writes, software   *
'*       clock-halt, software write-protection, trickle charge and                    *
'*       operation down to 2.0 volts @ 300 nA are other notable features.             *
'*****************************************************************************

'Define I/O pins and RTC variables
Clk        CON        0
Dta    CON    1
RTCCS      CON        2
RTCCmd     VAR        BYTE
Value      VAR        BYTE
Seconds VAR           BYTE
Minutes VAR           BYTE
Hours      VAR        BYTE
Date       VAR        BYTE
Month      VAR        BYTE
Day        VAR        BYTE
Year       VAR        BYTE
Idx        VAR        BYTE

'Define RTC Command Constants
SecReg     CON        %00000
MinReg  CON           %00001
HrsReg     CON        %00010
DateReg    CON        %00011
MonReg     CON        %00100
DayReg     CON        %00101
YrReg      CON        %00110
CtrlReg CON           %00111
TChgReg               CON        %01000
BrstReg    CON        %11111
```

```
'Define Days-Of-Week, Months and AM/PM text.
'All text is stored in EEPROM with a binary 0 as the end-of-text character
Sun      DATA "Sun",0
Mon      DATA "Mon",0
Tue      DATA "Tues",0
Wed      DATA "Wednes",0
Thu      DATA "Thurs",0
Fri      DATA "Fri",0
Sat      DATA "Satur",0

Jan      DATA "January",0
Feb      DATA "February",0
Mar      DATA "March",0
Apr      DATA "April",0
May      DATA "May",0
Jun      DATA "June",0
Jul      DATA "July",0
Aug      DATA "August",0
Sep      DATA "September",0
Oct      DATA "October",0
Nov      DATA "November",0
Dcm      DATA "December",0

AM       DATA " AM",0
PM       DATA " PM",0

'Set I/O pin states and directions
OUTS = %0000000000000000        'All logic low
DIRS = %0000000000000111        'I/O 0,1 and 2 are output, rest are input

Initialize:
 'Set Time and Date to 05/18/98 - 3:00 PM
 'NOTE: Date must be set only once for every power-up of DS1302 chip.
  Day     = $02                 'Monday
  Month   = $05                 'May
  Date    = $18        '18th
  Year    = $98        '1998
  Hours   = $15                 '3:00 PM (in 24-hour mode)
  Minutes = $00
  Seconds = $00
  GOSUB SetTimeAndDate

Loop:
  'Read out all date and time values and display them in two formats on
  'the debug screen.
  GOSUB ReadRTCBurst
  DEBUG HOME,"LONG FORMAT DATE AND TIME:",CR
  GOSUB PrintLongDate
  GOSUB Print12HourTime
  DEBUG CR,CR,"SHORT FORMAT DATE AND TIME:",CR
  GOSUB PrintShortDate
  GOSUB Print24HourTime
GOTO Loop

'=================== DS1302 Real-Time Clock Subroutines ===================

PrintLongDate:
 'Print long date format on debug screen
  LOOKUP Day-1,[Sun,Mon,Tue,Wed,Thu,Fri,Sat],Idx
  GOSUB PrintIt
  DEBUG "day, "
  LOOKUP Month-1,[Jan,Feb,Mar,Apr,May,Jun,Jul,Aug,Sep,Oct,Nov,Dcm],Idx
  GOSUB PrintIt
 'NOTE: The following line prints the proper 4-digit year for the years
 '1990 through 2089
  DEBUG " ",HEX2 Date,", ",DEC2 20-(Year/90),HEX2 Year, CR
RETURN

PrintShortDate:
  'Print short date format on debug screen
  DEBUG HEX2 Month,"/",HEX2 Date,"/",HEX2 Year, CR
RETURN

Print12HourTime:
 'Print 12-hour time format on debug screen
 'NOTE: The DS1302 has 12 and 24 hour time-keeping modes (bit 7 of HrsReg
 'sets 12/24 mode and bit 5 indicates AM/PM or 20+ hours).  For purposes
 'of this example, we're using 24 hour mode only, and converting it to
 '12-hour in the next two lines below.
  DEBUG DEC2 12-(24-(Hours.HIGHNIB*10+Hours.LOWNIB)//12),":",HEX2 Minutes,":",HEX2 Seconds
  LOOKUP Hours/$12,[AM,PM],Idx
  GOSUB PrintIt
RETURN

Print24HourTime:
 'Print 24-hour time format on debug screen
  DEBUG HEX2 Hours,":",HEX2 Minutes,":",HEX2 Seconds
RETURN

PrintIt:
  'Prints zero (0) terminated text from EEPROM
  READ Idx,Value                 'Get next character
  IF Value = 0 THEN Finished 'Make sure it's not a binary 0
  DEBUG Value                             'Display it on screen
  Idx = Idx + 1
  GOTO PrintIt
Finished:
RETURN
```

```
WriteRTCRAM:
 'Write to DS1302 RAM Register
  HIGH RTCCS
  SHIFTOUT Dta, Clk, LSBFIRST, [%0\1,RTCCmd\5,%11\2,Value]
  LOW RTCCS
RETURN

WriteRTC:
 'Write to DS1302
  HIGH RTCCS
  SHIFTOUT Dta, Clk, LSBFIRST, [%0\1,RTCCmd\5,%10\2,Value]
  LOW RTCCS
RETURN

ReadRTCBurst:
 'Read all time-keeping registers in one burst
  HIGH RTCCS
  SHIFTOUT DTA, Clk, LSBFIRST, [%1\1,BrstReg\5,%10\2]
  SHIFTIN DTA, Clk, LSBPRE, [Seconds,Minutes,Hours,Date,Month,Day,Year]
  LOW RTCCS
RETURN

ReadRTCRAM:
 'Read DS1302 RAM Register
  HIGH RTCCS
  SHIFTOUT DTA, Clk, LSBFIRST, [%1\1,RTCCmd\5,%11\2]
  SHIFTIN DTA, Clk, LSBPRE, [Value]
  LOW RTCCS
RETURN

SetTimeAndDate:
  'Write time values into all time-keeping registers, being sure to clear
  'the write-protect bit in CtrlReg before the write, and set the
  'write-protect bit after the write
  FOR Idx = 0 TO 8
    LOOKUP Idx,[0,Seconds,Minutes,Hours,Date,Month,Day,Year,128],Value
    LOOKUP Idx,[CtrlReg, SecReg, MinReg, HrsReg, DateReg, MonReg, DayReg, YrReg, CtrlReg],RTCCmd
    GOSUB WriteRTC
  NEXT
RETURN
```

BASIC Stamp II (BS2-IC) Program Listing #2

```
'****************************************************************************
'*      Title: DS1302_2.BS2    Author: Jeff A Martin    Date: 5/18/98        *
'*                                                                            *
'* Description: Shortened version of DS1302_1.BS2.                                   *
'*                                                                            *
'****************************************************************************


DATA (49)



RTCCmd          VAR     BYTE
Clk             CON     0
Dta     CON     1
RTCReset        CON     2
Temp    VAR     BYTE
Seconds VAR     BYTE
Minutes VAR     BYTE
Hours   VAR     BYTE
Date    VAR     BYTE
Month   VAR     BYTE
Year    VAR     BYTE
I       VAR     BYTE

'Define Constants
SecReg   CON    %00000
MinReg  CON     %00001
HrsReg   CON    %00010
DateReg  CON    %00011
MonReg  CON     %00100
YrReg    CON    %00110
CtrlReg CON     %00111
BrstReg  CON    %11111

DIRS = %0000000000111111
OUTS = %0000000000000000

'  Clear Write Protect bit in control register
  Temp = $10
  RTCCmd = CtrlReg
  GOSUB WriteRTC

  Temp = $98
  RTCCmd = YrReg
  GOSUB WriteRTC

  Temp = $08
  RTCCmd = MonReg
  GOSUB WriteRTC

  Temp = $27
  RTCCmd = DateReg
  GOSUB WriteRTC

  Temp = $48
  RTCCmd = MinReg
  GOSUB WriteRTC

  Temp = $00
  RTCCmd = SecReg
  GOSUB WriteRTC

  Temp = $80
  RTCCmd = CtrlReg
  GOSUB WriteRTC

Loop:
  GOSUB ReadRTCBurst
  DEBUG HOME,DEC Hours.HIGHNIB,DEC Hours.LOWNIB,":",DEC Minutes.HIGHNIB
  DEBUG DEC Minutes.LOWNIB,":",DEC Seconds.HIGHNIB,DEC Seconds.LOWNIB
  DEBUG " ",DEC Month.HIGHNIB,DEC Month.LOWNIB,"/"
  DEBUG DEC Date.HIGHNIB, DEC Date.LOWNIB,"/",DEC Year.HIGHNIB, DEC Year.LOWNIB,CR
GOTO Loop

WriteRTCRAM:
 'Write to DS1202 RTC
  HIGH RTCReset
  SHIFTOUT Dta, Clk, LSBFIRST, [%0\1,RTCCmd\5,%11\2,Temp]
  LOW RTCReset
RETURN

WriteRTC:
'Write to DS1202 RTC
  HIGH RTCReset
  SHIFTOUT Dta, Clk, LSBFIRST, [%0\1,RTCCmd\5,%10\2,Temp]
  LOW RTCReset
RETURN

ReadRTCBurst:
  HIGH RTCReset
  SHIFTOUT DTA, Clk, LSBFIRST, [%1\1,BrstReg\5,%10\2]
  SHIFTIN DTA, Clk, LSBPRE, [Seconds,Minutes,Hours,Date,Month,Year,Year]
  LOW RTCReset
RETURN

ReadRTCRAM:
  HIGH RTCReset
  SHIFTOUT DTA, Clk, LSBFIRST, [%1\1,RTCCmd\5,%11\2]
  SHIFTIN DTA, Clk, LSBPRE, [Temp]
  LOW RTCReset
RETURN
```