ZX-17 Serial Real-Time Clock application board

This application board is used for making the real-time clock data for any microcontroller system. It interfaces via a serial line. ZX-17 provides all time data included real-time clock, date, day in real calendar and square wave clock signal output in 1Hz, 4.096kHz, 8.192kHz and 32.768kHz selected by software.

Interfacing is very easy. It use only one line for serial data communication in half-duplex, TTL level, non-invert logic, 8-bit data, none parity, 1-bit STOP bit and baudrate 9,600 bit per second (9600 8-N-1). It is designed to suitable for any microcontroller included BASIC Stamp and PICAXE.

Application

- Data logger that need real-time clock data
- Real-time controller
- Programmable clock
- Entry system that need real-time clock data

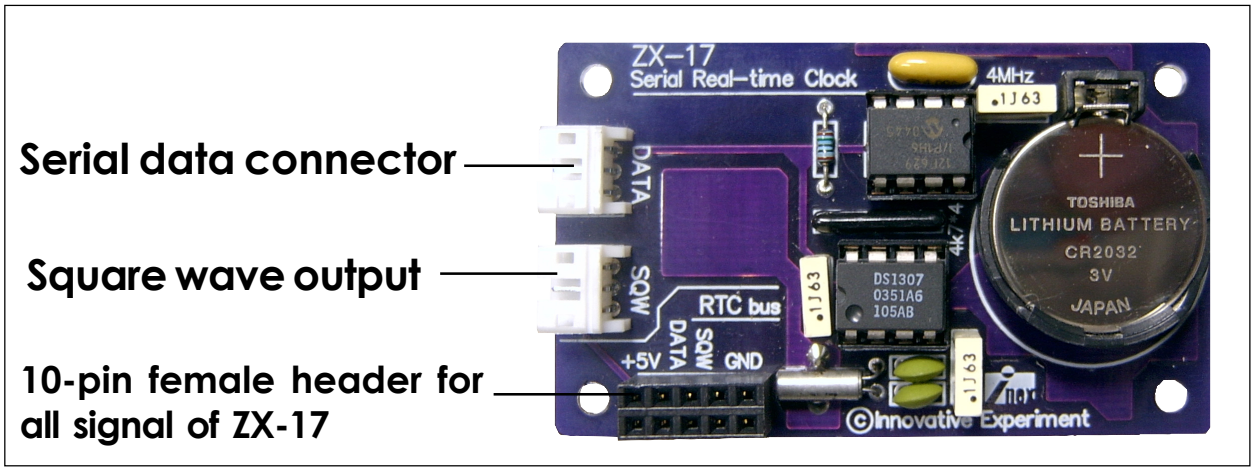


Figure 1 : Board layout of ZX-17 Serial Real-Time Clock application board

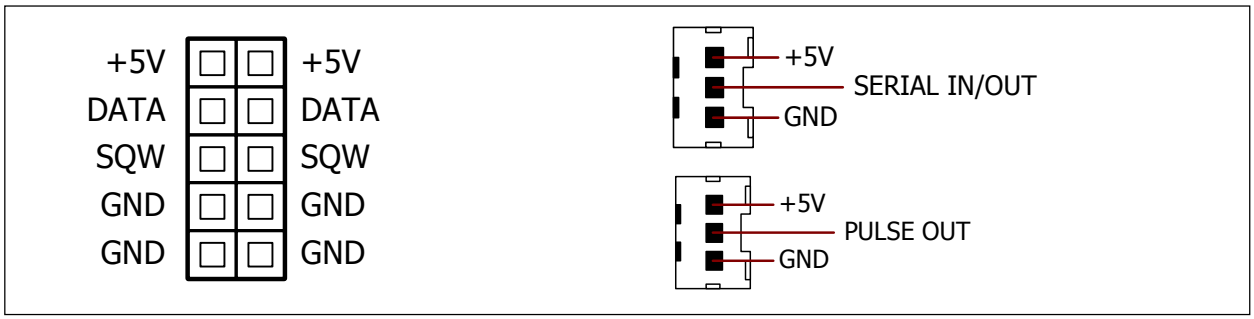


Figure 2 : Pin assignment of interface connector on ZX-17

Features

- Interface with any microcontroller included BASIC Stamp and PICAXE by serial data communication in half-duplex, TTL level, non-invert logic, 8-bit data, none parity, 1-bit STOP bit and baudrate 9,600 bit per second.
- Use Dallas's Real-time clock chip DS1307 with battery back-up for storing time data without supply voltage.
- 32.768kHz time-base clock frequency
- 2-interface connector for suitable application and experiment. One is 10-pin female header for SWG#22 hook-up wire and two 2mm. male PCB connector for data and square wave output
- Very easy in programming

Control command

1. Setting command

Use to set all time data parameter and select the square wave output

Format

```
S [0..7] [Value]
```

Detail

S0 : Second value setting following value [0-59]

S1 : Minute value setting following value [0-59]

S2 : Hour value setting following value [0-23]

S3 : Day value setting following value [1-7] for Monday [1], Tuseday [2], ..., Sunday [7]

S4 : Date value setting following value [1-31] depend on each month and year

S5 : Month value setting following value [1-12] mean January to December. It will show in 3 letters abbreviation (Jan to Dec)

S6 : Year value setting following value [0-99] for year 2000 to 2099

S7 : Set operation of Square wave output at SQWOUT following value as :

- 00 - set to send logic "0"
- 10 - 1Hz square wave
- 11 - 4.096kHz square wave
- 12 - 8.192kHz square wave
- 13 - 32.768 kHz square wave
- 80 - set to send logic "1"

2. Reading command

This command group is used for reading time data to store in Variable

Format

```
Value = R[0...7]
```

Detail

R0 : read Second value in range [0-59]

R1 : read Minute value in range [0-59]

R2 : read Hour value in range [0-23]

R3 : read Day value in range [1-7] for Monday to Sunday

R4 : read Date value in range [1-31] depend on each month and year

R5 : read Month value in range [1-12] mean January to December. It will show in 3 letters abbreviation (Jan to Dec)

R6 : read Year value in range [0-99] for year 2000 to 2099

RS7: read status of square wave at SQWOUT

- 00 - SQWOUT set to logic "0"
- 10 - SQWOUT set to 1Hz signal
- 11 - SQWOUT set to 4.096kHz signal
- 12 - SQWOUT set to 8.192kHz signal
- 13 - SQWOUT set to 32.768kHz signal
- 80 - SQWOUT set to logic "1"

3. Time frame data reading command

This command group is used for read all time data in full format.

t read time value in 9-byte ASCII code in 24 hours format include second value. This data has 8 byte as :

0	1	2	3	4	5	6	7	8
h	h	:	m	m	:	s	s	CR

`hh` is Hour value

`mm` is Minute value

`ss` is Second value

`CR` is Carrier Return code $0D or 13. Use to ending the data frame.

d read date data in short format. Display in 9-byte ASCII as :

0	1	2	3	4	5	6	7	8
D	D	/	M	M	/	Y	Y	CR

DD is Date value

MM is Month value

YY is Year value

CR is Carrier Return code $0D or 13. Use to ending the data frame.

T read time value in 12-byte ASCII code in 12 hours format include second value and AM/PM.

0	1	2	3	4	5	6	7	8	9	10	11
h	h	:	m	m	:	s	s		A/P	M	CR

hh is Hour value

mm is Minute value

ss is Second value

A/P and M is AM/PM time mode

CR is Carrier Return code $0D or 13. Use to ending the data frame.

D read date value in full format in 9-byte ASCII as :

0	1	2	3	4	5	6	7	8	9	10	11
D	D	-	Month			-	2	0	Y	Y	CR

DD is Date value

Month is Month value in 3-letters format

YY is Year value of last 2-digit

CR is Carrier Return code $0D or 13. Use to ending the data frame.

Example 1 Simple reading time value

Hardware : BASIC Stamp2SX board such as Stamp-BOX or SCi-BOX.

1.1 Connect ZX-17 to Stamp-BOX or SCi-BOX.

1.2 Write the listing program 1 into BASIC Stamp editor program.

1.3 Connect BASIC Stamp2SX hardware to computer's serial port.

1.4 Download this code to BASIC Stamp2SX board.

1.5 The Debug Terminal will appear and show the time data that store in ZX-17 board

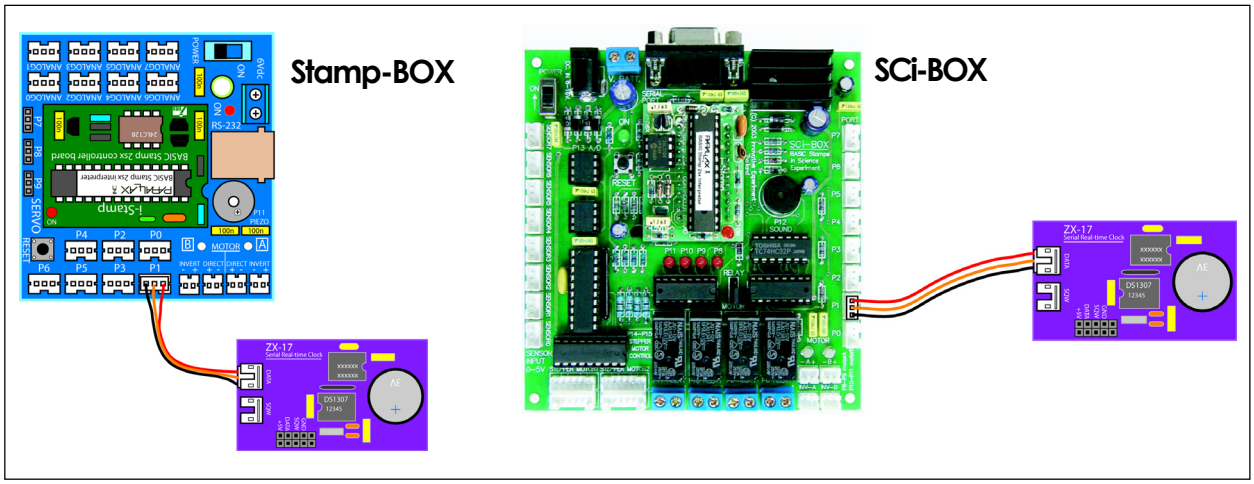
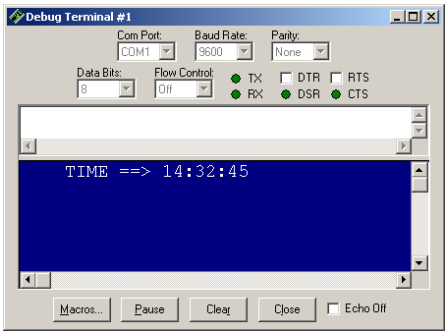


Figure 3 : interface diagram between some BASIC Stamp2SX board and ZX-17

```
' {$STAMP BS2sx}
' {$PBASIC 2.5}
' {$PORT COM1}

SECOND      VAR    Byte
MINUTE      VAR    Byte
HOUR        VAR    Byte

      PAUSE 1000
DO
      PAUSE 100
      SEROUT 1,240,["R",0]     ' Command For Read Second
      SERIN 1,240,[SECOND]     ' Save Second to Variable
      SEROUT 1,240,["R",1]     ' Command For Read Minute
      SERIN 1,240,[MINUTE]     ' Save Minute to Variable
      SEROUT 1,240,["R",2]     ' Command For Read Hour
      SERIN 1,240,[HOUR]            ' Save Minute to Variable
' Show All On Basic Stamp Debug Terminal
     DEBUG CRSRXY, 4, 5,"TIME ==> ",DEC2 HOUR,":",DEC2 MINUTE,":",DEC2 SECOND
LOOP
```



Listing 1 : Simple read time data from ZX-17 board with BASIC Stamp2SX

Example 2 Setting time

Hardware : BASIC Stamp2SX board such as Stamp-BOX or SCi-BOX.

From example 1, time data that read from ZX-17 may be not correct time. User can write the values to set the correct real-time value to ZX-17 application board.

2.1 Still connect ZX-17 to Stamp-BOX or SCi-BOX following figure 3

2.2 Write the listing 2 code into BASIC Stamp editor program

2.3 Connect BASIC Stamp2SX hardware to computer's serial port.

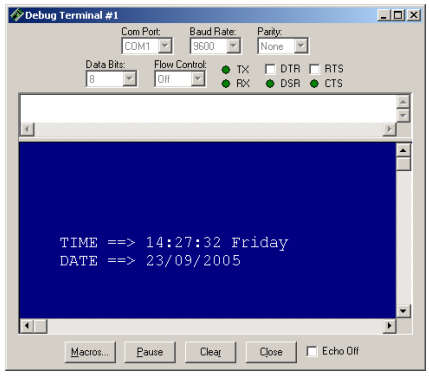
2.4 Download this code to BASIC Stamp2SX board.

2.5 The Debug Terminal will appear and show the time data that store in ZX-17.

```
' {$STAMP BS2sx}
' {$PBASIC 2.5}
' {$PORT COM1}

SECOND      VAR   Byte
MINUTE      VAR  Byte
HOUR        VAR  Byte
DAY         VAR  Byte
DATE        VAR  Byte
MONTHVAR  Byte
YEAR        VAR  Byte
STATUS      VAR  Byte

PAUSE 1000              ' First Time Delay For Wake up ZX-17
SEROUT 1,240,["S",0,20]' Set second = 20
SERIN 1,240,[STATUS]          ' Check Status and wait for write complete
'DEBUG CRSRXY,4,2 ,DEC2 STATUS          ' Show Status
SEROUT 1,240,["S",1,27]       ' set minute = 27
SERIN 1,240,[STATUS]                ' wait
SEROUT 1,240,["S",2,14]       ' set hour = 14
SERIN 1,240,[STATUS]
SEROUT 1,240,["S",3,05]       ' set day = friday
SERIN 1,240,[STATUS]
SEROUT 1,240,["S",4,23]       ' set date = 23
SERIN 1,240,[STATUS]
SEROUT 1,240,["S",5,09]       ' set month = september
SERIN 1,240,[STATUS]
SEROUT 1,240,["S",6,05]       ' set year = 2005
SERIN 1,240,[STATUS]
DO
' Show Time
     PAUSE 100
     SEROUT 1,240,["R",0]           ' Command For Read Second
     SERIN 1,240,[SECOND]           ' Save Second to Variable
     SEROUT 1,240,["R",1]           ' Command For Read Minute
     SERIN 1,240,[MINUTE]           ' Save Minute to Variable
     SEROUT 1,240,["R",2]           ' Command For Read Hour
     SERIN 1,240,[HOUR]             ' Save Minute to Variable
```



Listing 2 : Simple write time data for setting time of ZX-17 with BASIC Stamp2SX (continue)

```
' Show All On Basic Stamp Debug Terminal
     DEBUG CRSRXY, 4, 5,"TIME ==> ",DEC2 HOUR,":",DEC2 MINUTE,":",DEC2 SECOND
' Show Date
     SEROUT 1,240,["R",3]           ' Command For read Day
     SERIN 1,240,[DAY]        ' Save Day to Variable
     SEROUT 1,240,["R",4]           ' Command For read Date
     SERIN 1,240,[DATE]             ' Save Date to Variable
     SEROUT 1,240,["R",5]           ' Command For read Month
     SERIN 1,240,[MONTH]            ' Save Month to Variable
     SEROUT 1,240,["R",6]           ' Command For Read Year
     SERIN 1,240,[YEAR]             ' Save Year to Variable
' Show All On Basic Stamp Debug Terminal
     DEBUG CRSRXY,4,6,"DATE ==> ",DEC2 DATE,"/",DEC2 MONTH,"/","20",DEC2 YEAR
' Show day in text
     SELECT DAY
     CASE 1
           DEBUG CRSRXY, 22, 5,"Monday"
     CASE 2
           DEBUG CRSRXY, 22, 5,"Tuesday"
     CASE 3
           DEBUG CRSRXY, 22, 5,"Wednesday"
     CASE 4
           DEBUG CRSRXY, 22, 5,"Thusday"
     CASE 5
           DEBUG CRSRXY, 22, 5,"Friday"
     CASE 6
           DEBUG CRSRXY, 22, 5,"Saturday"
     CASE 7
           DEBUG CRSRXY, 22, 5,"Sunday"
     ENDSELECT
LOOP
```

Listing 2 : Simple write time data for setting time of ZX-17 with BASIC Stamp2SX (finish)

How to check the writing operation :

In time setting, after write value one byte must check before write the next byte. See the program 2, after send command

```
SEROUT 1,240,["S",0,20]
```

to set second value to 20 must following to send this command

```
SERIN 1,240,[STATUS]
```

to check the writing status. After writing value finish, writing value will read to store into STTUS variable. Use can check by insert command

```
DEBUG CRSRXY,4,2,DEC2 STATUS
```

to open the Debug Terminal for displaying data that written to STATUS variable. The Debug Terminal must display value 20 on screen.

Displaying value

Normally, result displaying will show both value include time and date. The operation will read time value and display before. After that read the date value to display. About day value will show in number 1 to 7 then must convert to simple word (Monday to Sunday) that understand by `SELECT CASE` command.

Example 3 Show time data to Serial LCD

Hardware :

1. BASIC Stamp2SX board such as Stamp-BOX or SCi-BOX.
2. ZX-17 Serial Real-time Clock appication board
3. SLCD16x2 Serial LCD 16-characters 2-lines

In this example, show read data from ZX-17 in time data frame by one command and show the value on Serial LCD.

3.1 Connect ZX-17 and Serial LCD to Stamp-BOX or SCi-BOX following figure 4

3.2 Write the listing program 3 into BASIC Stamp editor program

3.3 Download this code to BASIC Stamp2SX board and see the operation form Serial LCD.

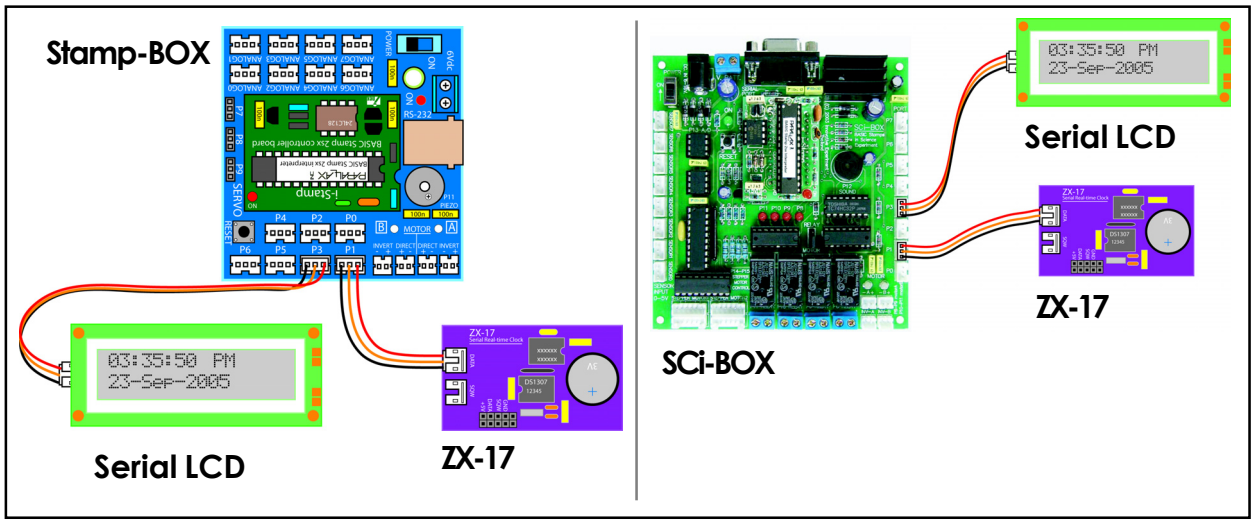


Figure 4 : Interface diagram of some BASIC Stamp2SX board, ZX-17 and Serial LCD

```
' {$STAMP BS2sx}
' {$PBASIC 2.5}
SAVE    VAR     Byte(13)        ' Reserve 13 byte array space
RTC     PIN     1               ' ZX-17 data pin
LCD     PIN     3               ' serial LCD data pin
PAUSE 1000
DO
   SEROUT RTC,240,["D"]           ' Send command for DATE full format
   SERIN    RTC,240,[STR SAVE\12]    ' Save 12 Byte data to memory
   SEROUT LCD,240,[$FE,$80,STR SAVE\11]
      ' show on serial LCD module;last byte ignore
   PAUSE 100
   SEROUT RTC,240,["T"]           ' Send command for TIME full format
   SERIN RTC,240,[STR SAVE\12]    ' Save 12 Byte data to memory
   SEROUT LCD,240,[$FE,$C0,STR SAVE\11]
      ' show on serial LCD module ; last byte ignore
   PAUSE 100
LOOP
```

Listing 3 : Read time data from ZX-17 in full format data frame with BASIC Stamp2SX

Example 4 Using SQW output

At SQW output of ZX-17 application board can send the square wave output with 4 values 1Hz, 4.096kHz, 8.192kHz and 32.768kHz. User can use this signal to time based for other application. Because the signal output's frequency is very accuracy.

Hardware : BASIC Stamp2SX board such as Stamp-BOX or SCi-BOX.

4.1 Connect ZX-17 to Stamp-BOX or SCi-BOX following figure 5

4.2 Write the listing program 4 into BASIC Stamp editor program.

4.3 Connect BASIC Stamp2SX hardware to computer's serial port.

4.4 Download this code to BASIC Stamp2SX board.

This example will show the reading status of SQW pin of DS1307 and drives sound to piezo speaker on the hardware. If use SCi-BOX board, must check the BASIC Stamp2SX pin in FREQOUT command to 12.

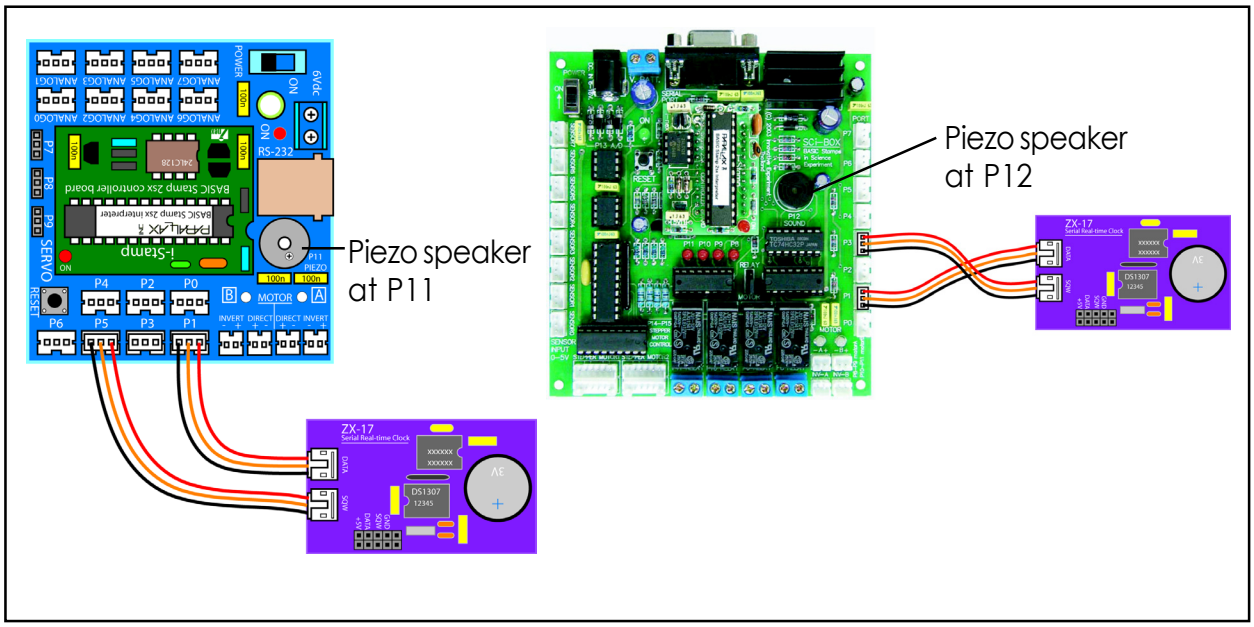


Figure 5 : SQW output interface to testing with BASIC Stamp2SX board

```
'  {$STAMP BS2sx}
'  {$PBASIC 2.5}

SEROUT  1,240,["S",7,10]                ' set 1Hz
DO
    IF IN5=0 THEN
      FREQOUT  11,200,2000              ' Sound
WAITS:  IF IN5 = 0 THEN GOTO WAITS      ' Wait until SQW = 1
    ENDIF
LOOP
```

Listing 4 : Test SQW output function of ZX-17 listing with BASIC Stamp2SX

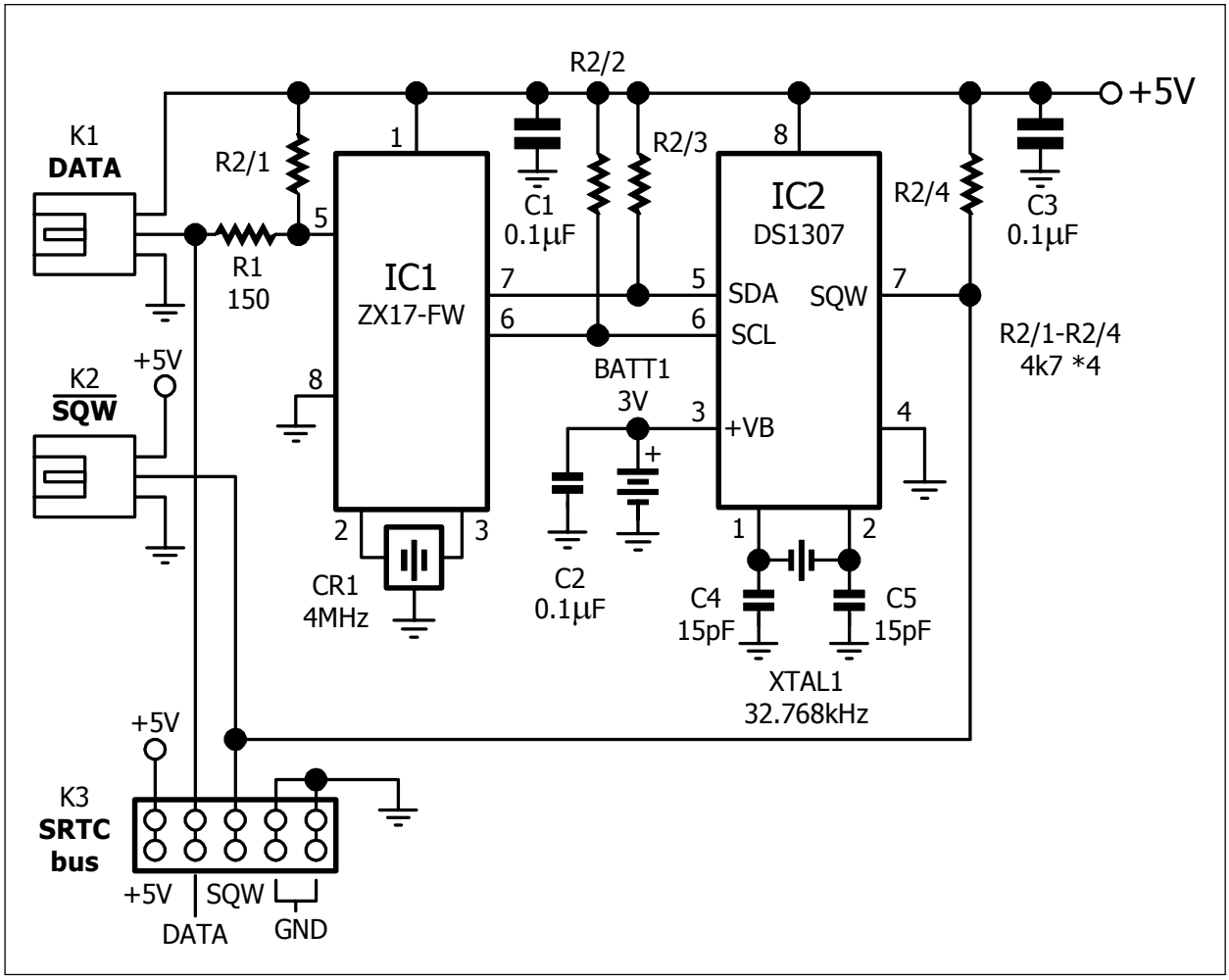


Figure 6 : ZX-17 Serial Real-Time Clock application board schematic diagram

ZX-17 board : Copyright 2005 Innovative Experiment Co.,Ltd. Thailand.