599 Menlo Drive, Suite 100
Rocklin, California 95765, USA
Office: (916) 624-8333
Fax: (916) 624-8003

General: info@parallax.com
Technical: support@parallax.com
Web Site: www.parallax.com
Educational: www.stampsinclass.com

DS2760 Thermocouple Kit (#28022)

1-Wire® Thermocouple Interface

Introduction

Thermocouples provide a low-cost, reliable means of measuring temperature over a wide range. The challenge when using a thermocouple is accurately measuring the very low Seebeck output voltage (fractional to low millivolts) from the element, and providing for cold junction temperature compensation.

The Dallas/Maxim DS2760 High Precision Li+ Battery Monitor is very easily configured into an effective thermocouple interface. The Parallax DS2760 Thermocouple Module capitalizes on this application and provides a complete connection between the BASIC Stamp and a standard thermocouple element.

Features

- 1-Wire® interface allows multiple devices with just one Stamp IO pin
- Cold Junction measurement: 0°C to +127°C (0.125°C resolution)
- Low power consumption:
  -- Active current: 90 μA max
  -- Sleep current: 2 μA max

Packing List

Verify that your DS2760 kit is complete in accordance with the list below:

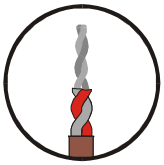
- DS2760 Thermocouple Module #550-28022
- (3) Thermocouple elements:
  -- (1) K-type (Chromel / Alumel) #800-00011
  -- (1) J-type (Iron / Constantan) #800-00012
  -- (1) T-type (Copper / Constantan) #800-00010
- This documentation

Note: DS2760 demonstration software may be downloaded from www.parallax.com.

Connections

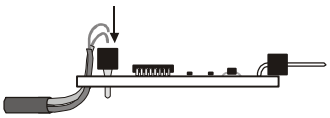
Before connecting the DS2760 Thermocouple Module to the BASIC Stamp you will need to prepare a thermocouple element, and then connect it to the cold junction port of the module. Start by carefully removing about one inch (250 mm) of the outer sleeve from each end of the element. From each lead on the temperature measurement end, remove about ½ inch (125 mm) of insulation and then carefully twist together (using pliers if necessary) and trim as shown in Figure 1.

Figure 1: Thermocouple Junction



On the cold junction (DS2760 module) end of the element, remove only ¼ inch (60 mm) of insulation from each lead. Route these leads through the bottom of the thermocouple module PCB and insert snuggly into the pin sockets as shown in Figure 2.

Figure 2: Cold Junction Connection to DS2760 PCB

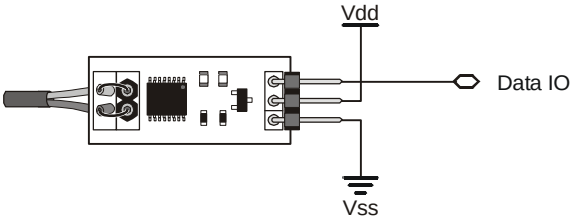


Use this table to ensure that you make the proper thermocouple connections to the module. If the leads are reversed, the measured temperature will be incorrect.

Type	Materials	SNS	Vss
K	Chromel / Alumel	Red	Yellow
J	Iron / Constantan	Red	White
T	Copper / Constantan	Red	Blue

Finally, the DS2760 Thermocouple Module is connected to the BASIC Stamp as shown in Figure 3 below (Note that the module includes a 4.7 KΩ pull-up on the 1-Wire® data line).

Figure 3: DS2760 Connections to BASIC Stamp



BASIC Stamp Application

The following BASIC Stamp application will run on either the BS2p or BS2pe and demonstrates how easy measuring wide-range temperatures can be when using the DS2760 Thermocouple Module. Other Stamps will require a Serial-to-1-Wire protocol converter, as well as code to manage the large tables across program slots, and are not covered in this document.

A little background: When two dissimilar metal wires are joined, a voltage will be developed across the open end that is proportional to the temperature difference between the joined and open ends. This effect was discovered by Thomas Seebeck in 1821. Through empirical testing, voltage tables have been established that correspond to the thermocouple junction temperature. These tables, however, use a cold junction (voltage measurement point) reference of zero degrees Celsius, forcing electronic devices to employ cold junction compensation.

Using the DS2760 we can measure the Seebeck voltage from the thermocouple with a resolution of 15.625 microvolts, then measure the cold junction temperature with a resolution of 0.125 degrees Celsius. A simple table look-up using the cold junction temperature will give us the cold junction compensation voltage. This is combined with the Seebeck voltage and, using a modified binary search algorithm, we can determine the compensated temperature from the thermocouple data table.

```
' =========================================================================
'
'   File...... DS2760TC_Demo.BPE
'   Purpose... Thermocouple temperature measurement using the DS2760
'   Author.... Parallax, Inc.  (Copyright 2004, All Rights Reserved)
'   E-mail.... support@parallax.com
'   Started...
'   Updated... 19 JAN 2004
'
'   {$STAMP BS2pe, KTablePos.BPE, JTablePos.BPE, TTablePos.BPE}
'   {$PBASIC 2.5}
'
' =========================================================================


' -----[ Program Description ]---------------------------------------------
'
' This program lets a BS2p or BS2pe read the temperature from the Parallax
' DS2760 thermocouple module.  User input of thermocouple type (K, J, or T)
' and temperature display is via the DEBUG window.


' -----[ Revision History ]------------------------------------------------


' -----[ I/O Definitions ]-------------------------------------------------

OW              PIN     8                       ' 1-Wire buss pin
```

```
' -----[ Constants ]-------------------------------------------------------

ReadNet         CON     $33                     ' read OW net address
SkipNet         CON     $CC                     ' skip OW net address
RdReg           CON     $69                     ' read register


' -----[ Variables ]-------------------------------------------------------

idx             VAR     Nib                     ' loop counter
type            VAR     Nib                     ' device type
char            VAR     Byte                    ' display byte/char

vIn             VAR     Word                    ' in millivolts
tmpCJ           VAR     Word                    ' device temp in C
tCuV            VAR     Word                    ' thermocouple millivolts
sign            VAR     Word                    ' TC sign bit

cjComp          VAR     Word                    ' temp compensation
tempC           VAR     Word                    ' temp in Celsius
tempF           VAR     Word                    ' temp in Fahrenheit

tblLo           VAR     Word                    ' table pointers
tblHi           VAR     Word
eePntr          VAR     Word
testVal         VAR     Word                    ' test value from table
error           VAR     Bit                     ' 1 = out of range


' -----[ EEPROM Data ]-----------------------------------------------------


' -----[ Initialization ]---------------------------------------------------

Stamp_Check:
  #IF ($stamp < BS2P) #THEN
    #ERROR "This program requires BS2p or BS2pe"
  #ENDIF

Check_Device:
  OWOUT OW, %0001, [ReadNet]                    ' get serial number
  OWIN  OW, %0010, [SPSTR 8]                    ' store in SPRAM
  GET idx, char                                 ' read device type
  IF (char <> $30) THEN                         ' if not $30, wrong device
    DEBUG "No DS2760 found."
    STOP                                        ' stop program
  ENDIF
```

```
Menu:
  DEBUG CLS,
        "================================", CR,
        " DS2760 Thermocouple Interface ", CR,
        "================================", CR,
        CR,
        "Select TC Type (1 - 3)", CR,
        CR,
        "(1) K - Chromel/Alumel", CR,
        "(2) J - Iron/Constantan", CR,
        "(3) T - Copper/Constantan", CR,
        CR,
        ">>> "

  DEBUGIN DEC1 type                                ' get selection
  IF (type < 1) OR (type > 3) THEN Menu            ' validate selection
  DEBUG CRSRXY, 0, 3, CLRDN                        ' remove selections
  STORE type                                       ' point READ to table

Show_SN:
  DEBUG CRSRXY, 0, 4, "Device SN... "
  FOR idx = 0 TO 7
    GET idx, char
    DEBUG HEX2 char
  NEXT

Show_Type:
  DEBUG CRSRXY, 0, 6, "TC Type..... "
  LOOKUP (type - 1), ["KJT"], char
  DEBUG char


' -----[ Program Code ]----------------------------------------------------

Main:
  DO
    GOSUB Read_TC_Volts                            ' read Seebeck voltage
    GOSUB Read_CJ_Temp                             ' read cold junction temp
    READ (tmpCJ * 2), Word cjComp                  ' get compensation voltage

    ' combine cjComp and tCuV
    '
    IF sign THEN
      ' TC below cold junction
      IF (tCuV < cjComp) THEN
        cjComp = cjComp - tCuV
      ELSE
        cjComp = 0                                 ' limit to 0C
      ENDIF
```

```
    ELSE
      ' TC above cold junction
      cjComp = cjComp + tCuV
    ENDIF

    LOOKUP type, [1023, 1023, 400], tblHi       ' set high end of search
    GOSUB TC_Lookup                             ' reverse lookup of table
    tempF = tempC * 9 / 5 + 32                  ' x 1.8 + 32

    IF (error = 0) THEN
      DEBUG CRSRXY, 0, 7,
            "Temp °C..... ", SDEC tempC, CLREOL
      DEBUG CRSRXY, 0, 8,
            "Temp °F..... ", SDEC tempF, CLREOL
    ELSE
      DEBUG CRSRXY, 0, 7,
            "Temp °C..... Out of Range", CLREOL
      DEBUG CRSRXY, 0, 8,
            "Temp °F..... Out of Range", CLREOL
    ENDIF

    PAUSE 1000
  LOOP
  END


' -----[ Subroutines ]-----------------------------------------------------

' Reads device input voltage (Vin pin)
' -- mV in millivolts (max reading is 4.75 volts)

Read_Vin:
  OWOUT OW, %0001, [SkipNet, RdReg, $0C]
  OWIN  OW, %0010, [vIn.BYTE1, vIn.BYTE0]
  IF (vIn.BIT15) THEN                           ' check sign
    vIn = 0                                     ' disallow negative
  ELSE
    vIn = vIn >> 5 */ $4E1                      ' x 4.88 millivolts
  ENDIF
  RETURN


' Reads current register to get TC voltage
' -- each raw bit = 15.625 uV
' -- tCuV in microvolts

Read_TC_Volts:
  OWOUT OW, %0001, [SkipNet, RdReg, $0E]        ' read current register
  OWIN  OW, %0010, [tCuV.BYTE1, tCuV.BYTE0]
```

```
  sign = tCuV.BIT15                                 ' save sign bit
  tCuV = tCuV >> 3                                  ' correct alignment
  IF sign THEN
    tCuV = tCuV | $F000                             ' pad 2's-compliment bits
  ENDIF
  tCuV = ABS tCuV */ 4000                           ' x 15.625 uV
  RETURN


' Reads cold junction (device) temperature
' -- each raw bit = 0.125 degrees C
' -- returns tmpCJ in whole degrees C

Read_CJ_Temp:
  OWOUT OW, %0001, [SkipNet, RdReg, $18]
  OWIN  OW, %0010, [tmpCJ.BYTE1, tmpCJ.BYTE0]
  IF (tmpCJ.BIT15) THEN                             ' check sign
    tmpCJ = 0                                       ' disallow negative
  ELSE
    tmpCJ = tmpCJ.HIGHBYTE                          ' >> 5 x 0.125 (>> 3)
  ENDIF
  RETURN


' Search currently selected TC table for nearest entry
' -- uses modified binary algorithm to find cjComp
' -- high end of search set before calling (tblHi)
' -- successful search sets tempC

TC_Lookup:
  tblLo = 0                                         ' low entry of table
  tempC = 22                                        ' default to room temp

  READ (tblHi * 2), Word testVal                    ' check max temp
  IF (cjComp > testVal) THEN
    error = 1                                       ' out of range
  ELSE
    DO
      eePntr = (tblLo + tblHi) / 2                  ' midpoint of search span
      READ (eePntr * 2), Word testVal               ' read value from midpoint

      IF (cjComp = testVal) THEN
        EXIT                                        ' found it!
      ELSEIF (cjComp < testVal) THEN
        tblHi = eePntr                              ' search lower half
      ELSE
        tblLo = eePntr                              ' search upper half
      ENDIF
```

```
    IF ((tblHi - tblLo) < 2) THEN             ' span at minimum
      eePntr = tblLo
      EXIT
    ENDIF
  LOOP
  tempC = eePntr
ENDIF
RETURN
```

Additional Resources

- Advanced thermocouple interface software (download from Parallax)
- Web Links:
  -- www.maxim-ic.com/quick_view2.cfm/qv_pk/2931
  -- www.capgo.com/Resources/Sensors/Temperature/Thermocouple/Thermocouple.html
  -- instserv.com/rmocoupl.htm
  -- instrumentation-central.com/pages/thermocouple_reference_table.htm

DS2760 Module Schematic