

SER1CH-UA User Manual

Symmetric Research
www.symres.com

SR document nnnnn.rrr
Copyright © 2007

User Manual Rev (2007/07/10)
Board Rev B

FREE WEB VERSION - PARTIAL CIRCUIT DIAGRAMS

Contents

1	Introduction	6
2	Installing the SER1CH-UA	7
2.1	Hardware installation	7
2.2	Software installation	9
2.3	Do's and Don'ts	10
3	Software	11
3.1	DVM: A Digital Volt Meter program	11
3.2	Setting parameters to control DVM (DvmSetup.ini)	13
3.3	Output file format for DVM (DvmOutput.asc)	13
3.4	Modifying DVM	13
3.5	CAL: A calibration program	15
3.6	Modifying CAL	18
3.7	Determining the correct serial port	19
3.8	SER1CH DLL Library	19
3.8.1	Open device	21
3.8.2	Read data	22
3.8.3	Scale data	23
3.8.4	Get error message string	25
3.8.5	Close device	26
3.8.6	Code example	27
4	Sigma Delta A/D converters	28
5	60Hz and 50Hz power line rejection	29
6	Sampling rate	30
7	Resolution	31
8	Ratiometric measurements	33
9	Scaling active sensor voltages	36
9.1	Resistive scaling and biasing	37
9.2	Op Amp scaling and biasing	39

10 Example applications	41
10.1 Basic voltage measurements with probes	42
10.2 Plotting results with GnuPlot	45
10.3 Ratiometric potentiometer	48
10.4 Scaling and biasing +/-10 volt signals into (0,5)	51
10.5 Measuring temperature	53
11 Absolute DC calibration	55
12 Single ended grounding practices	56
13 Batteries and minimizing power	57
13.1 Measuring current consumption	58
13.2 Minimizing power	59
13.3 Batteries	60
14 Frequently Asked Questions	61
14.1 Software	62
14.2 Hardware	62
15 Specifications	67
16 Circuit diagrams	68
17 Extra supplies	71
17.1 Small parts for cables etc	72

List of Figures

2.1	VIN connected to VREF	8
2.2	VIN connected to GND	9
2.3	VIN connected to VREF	9
3.1	Screen shot of GUI version of DVM program	12
3.2	Sample DVM .ini initialization file	14
3.3	DVM .ini keyword table	15
3.4	Sample DvmOutput.asc output file in verbose mode.	16
3.5	Sample DvmOutputBare.asc output file in bare mode.	17
3.6	Screen shot of GUI version of CAL program	18
3.7	Ser1ch library function table	20
3.8	Sample program calling Ser1ch library functions	27
7.1	SER1CH-UA long term resolution plot	31
8.1	Potentiometer parameters	34
9.1	Two resistor scaling and biasing (-5,+5) into (0,5)	37
9.2	Three resistor scaling and biasing (-10,+10) into (0,5)	38
9.3	Three resistor values for various input ranges with $V_{bias} = VREF$	38
9.4	Op amp unity gain buffer with scaling	39
9.5	Op amp buffer with gain of 10	40
10.1	Two lead voltage measurement of a battery	42
10.2	Pomona test leads for voltage measurements	43
10.3	Using a solar cell as a light level meter	44
10.4	DVM output file fragment from DvmOutputSolar.asc	45
10.5	DVM startup file fragment from DvmSetupSolar.ini	46
10.6	GnuPlot commands for plotting DvmOutputSolar.asc	46
10.7	SER1CH-UA solar cell data example plot	47
10.8	10 turn potentiometer, SER1CH-UA connections	48
10.9	10 turn potentiometer, reverse side connections	49
10.10	Linear slider potentiometer	50
10.11	Scaling and biasing (-10,+10) with four 10K resistors	51
10.12	Temp diode with simple resistive current source	53
10.13	Temp diode with constant current op amp source	54
13.1	Maximum base power consumption	58

15.1 Specifications 67

Chapter 1

Introduction

The Symmetric Research SER1CH-UA is a precision analog to digital converter for DC applications requiring 24 bit A/D conversions at a 1Hz sampling rate.

A leading feature of the system is its low power consumption while still providing excellent resolution. With an active power requirement of 600 microamperes at 9 volts, the system can be easily powered from batteries or solar cells for field applications.

Besides having an analog input for A/D conversions, a buffered copy of the A/D reference is also provided for the user. This is important for ratiometric applications, such as measuring potentiometer positions and other sensor values. By using the same reference voltage for the A/D and the sensor, temperature drift is reduced.

The digital interface of the system is to an RS232 serial port. Software access is from user space without the need for a kernel mode device driver. A Windows GUI DVM application is included for displaying conversion results as well as saving acquired data to a hard disk.

Easy to use, the system requires power from a wall transformer or battery and connecting the user's sensor to the input banana jacks. Among the features of the SER1CH-UA are:

- Low power consumption, 600 microamperes during conversion
- Power supply voltage can range from 9 to 18 volts
- 24 bit A/D converter design
- Buffered analog input with 10M ohm input impedance
- Buffered A/D reference voltage provided for ratiometric measurements
- General use includes applications with potentiometers and other passive sensors
- Banana input plugs, test leads, wall transformer, and software included

We hope the SER1CH-UA is a useful tool for your applications

Chapter 2

Installing the SER1CH-UA

Installation of the SER1CH-UA is straightforward. The following steps should get you up and running quickly.

2.1 Hardware installation

Connect the SER1CH-UA to the supplied wall transformer with the 2.1mm power connector. The green power LED on the top panel should light up. If not, make sure the wall transformer has power.

Connect the RS232 DB9 connector on the back panel of the SER1CH-UA to your PC with the supplied cable. Make a note of which COM port you are using on your computer. You will need to know the COM number when running the application software.

Analog connections are made to the three banana jacks on the top panel of the SER1CH-UA. Reading from left to right, the jacks are (GND, VIN, VREF) with the colors (black, red, red) as shown in Figure 2.1.

The middle red VIN banana jack is the analog input. All voltages are measured with respect to GND, the black banana jack. When you are connecting an external voltage use the VIN and GND banana jacks to connect the voltage, keeping in mind the GND jack is the same as PC system ground.

The red banana jack on the outer right is the VREF output jack. Don't connect the VREF jack directly to ground. It is not fatal, but will draw unnecessary current.

For initial testing connect the VIN jack to the GND or VREF jacks to place known voltages on the input as in Figure 2.1. When connecting VIN to VREF, you don't need to make a connection to GND because the VIN and VREF share the same ground internally.

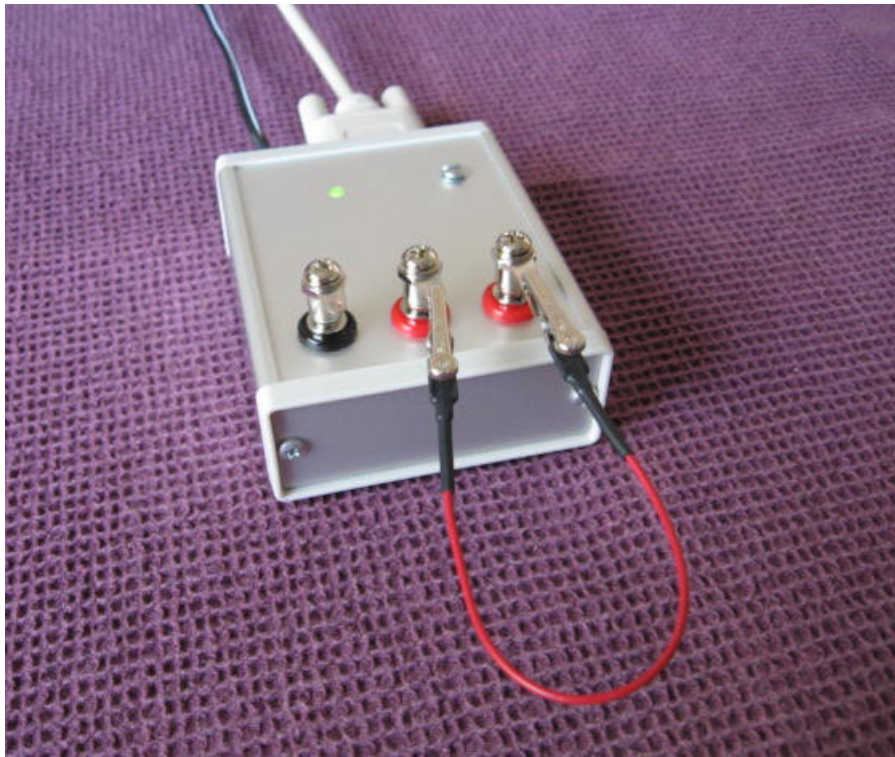


Figure 2.1: VIN connected to VREF



Figure 2.2: VIN connected to GND



Figure 2.3: VIN connected to VREF

2.2 Software installation

Run the install.bat file on the CD. The directory /SR/SER1CH-UA will be created on your hard disk and files will be copied there. No changes will be made to the operating system registry or any other system files.

The SER1CH-UA does not require a device driver or device driver installation. Application programs such as DvmGui.exe may be run as soon as the SER1CH-UA is connected to a serial port and the SER1CH-UA is powered up.

To uninstall the software, simply delete the /SR/SER1CH-UA directory.

2.3 Do's and Don'ts

- **DO** use the provided 9 vdc wall transformer or a 9 volt battery to power the system.
 - **DO** use the system with RS232 ports that support RTS, DTR, and CTS.
 - **DO** read the *FAQ* chapter for general questions regarding the SER1CH-UA.
 - **DO** use the SER1CH-UA to measure positive voltages in the (0,5) volt range on the red VIN banana jack with respect to the black GND banana jack.
-
- **DON'T** exceed 18 volts on the power supply. Excessive power supply voltages will result in high power supply current drain.
 - **DON'T** expect to make precision measurements with floating inputs. The analog input VIN must be connected to a voltage source to make an accurate measurement. VREF and GND are convenient voltages for initial testing.
 - **DON'T** short circuit the VREF A/D reference output banana jack to the black GND jack. The output is current limited so no damage will occur, however any current sourced by VREF is ultimately provided by the SER1CH-UA power supply.
 - **DON'T** use the SER1CH-UA to measure 110 vac or other high voltages. You will damage the system and perhaps even your computer.

Chapter 3

Software

The SER1CH-UA comes with some ready-to-run application programs to help you begin acquiring data right away.

Installing the SER1CH-UA software creates a directory called /SR/SER1CH-UA. The DVM subdirectory contains the main acquisition program while the Calibrate subdirectory contains a calibration program to help prepare the initialization file used by DVM. These programs are described below.

3.1 DVM: A Digital Volt Meter program

This program uses the SER1CH-UA to acquire data which it displays on the screen and saves to a file. There are two versions of DVM. DvmCmd.exe is a console program that runs from the command line and DvmGui.exe is a Windows Graphical User Interface (GUI) program that uses the Microsoft Foundation Classes (MFC) libraries.

Both versions of DVM read control settings from an initialization file such as DvmSetup.ini which is described in more detail in the next section. Both versions also output the acquired data to an ASCII file such as DvmOutput.asc. The format of this file is described in section 3.3 below.

Running either version of DVM is simple. From Windows Explorer, just double click on the RunDvmCmd or RunDvmGui shortcut to start DvmCmd.exe or DvmGui.exe respectively. The RunDvmCmd shortcut first starts a command prompt and then runs DvmCmd.exe. You may also type in at a command prompt either program name followed by the name of the .ini file you wish to use.

If no .ini file is specified, DVM will output a .ini file named DvmSetupDefault.ini containing the default settings. You should copy this file to a new name, modify it to have the settings you prefer and run DVM again with the new .ini file as a command line or shortcut argument.

Once DvmCmd.exe starts, it displays a prompt showing the serial port it will use. If this is not correct, please quit, edit the .ini file and try again. Also, see section 3.7 on page 19 for a brief discussion about how to choose a valid serial port. Once the serial port selection is correct, press the ENTER key to begin acquiring data.

Once DvmGui.exe starts, it displays a dialog box showing the default serial port specified in the .ini file. If this is not correct, use the serial port dropdown list box to select the serial port that *is* connected to the SER1CH-UA. Then click the run button to begin acquiring data.

If the RunMode Single radio button is checked, a single data sample will be acquired each time the run button is pressed. Otherwise, DvmGui will acquire data samples once a second until the stop button is pressed.

DVM will not overwrite an existing .asc output file. So, if you've run DvmGui or DvmCmd before, you may need to rename or delete the existing output file before DVM will start running. Changing the OutputFileName value in the .ini file will also work. Although you must stop and restart DvmGui before it will read in the new value.

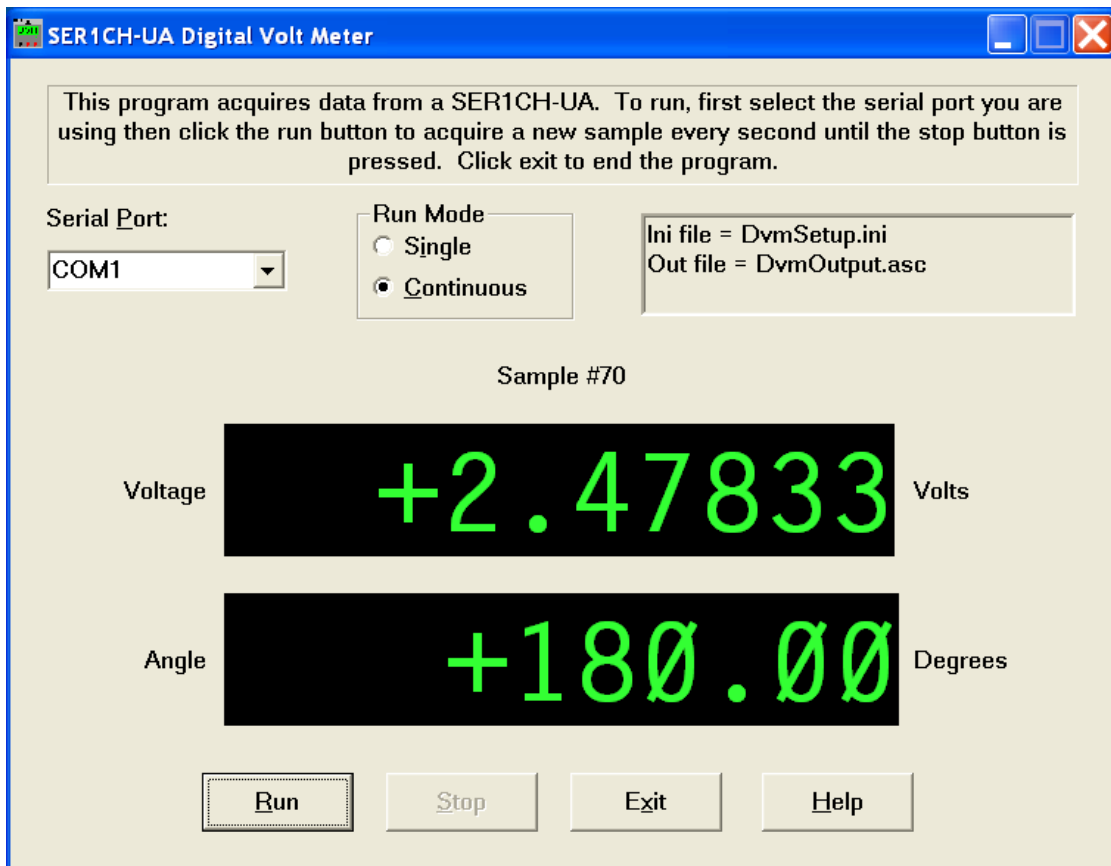


Figure 3.1: Screen shot of GUI version of DVM program

Now you should see the data acquired from the single SER1CH-UA channel being displayed twice on the dialog box. These two displays represent the same data scaled into different units. For example, the first display could show volts while the second display could show user units such as inches or degrees or ounces or etc depending on the specific type of sensor being used. Both displays are generated from the raw data, which comes in as A/D counts, by applying a simple linear scale

and offset transformation.

This transformation is really nothing more than using the familiar $y = mx + b$ equation for a straight line where the scale factor m and the offset b are controlled by user settable calibration parameters located in the DVM .ini initialization file. The easiest way to determine the proper parameter values is to use the provided calibration program described in section 3.5 on page 15 below.

3.2 Setting parameters to control DVM (DvmSetup.ini)

Control settings for DVM are specified in the DVM .ini initialization file. This section describes the various parameters in more detail below. But first let's see what a DVM .ini file looks like. Several additional sample .ini files are included in the /SR/SER1CH-UA/DVM directory.

The DVM .ini format is based on a 'keyword = value' structure where each line contains a keyword, an equal sign, and the value. Blank lines are ignored and semi-colons start comments which are also ignored. Lines of the form 'keyword number = value' are also allowed for certain keywords which apply to parameters that may be repeated for different instances such as the two displays of the data.

The following table describes all the keywords.

3.3 Output file format for DVM (DvmOutput.asc)

Both versions of DVM save the acquired data to an ASCII output file. Typically it is named DvmOutput.asc, but you can select the filename with the OutputFileName keyword in the DVM .ini file. The OutputFileShowHeader keyword controls whether the output file includes header information or only the data. When generating data files to import into Excel, use OutputFileShowHeader=OFF to suppress the header text and include only the data values themselves.

Now let's see what some DVM .asc output files look like. The first sample output file was created in verbose mode with all the optional outputs turned on. It has OutputFileShowHeader=ON so the header information is included and has OutputFileShowIndex=ON so an extra column with the sample number is included. In addition, it has OutputFileShowTimeSec=ON and OutputFileShowTimeYmd=ON which add columns containing time as seconds since 1970 and as year/month/day respectively.

The second sample DVM .asc output file was created in bare mode with all the optional outputs turned off so only the data values are included. This mode makes it easy for the file to be imported into analysis programs like Excel or MatLab.

3.4 Modifying DVM

For those who are comfortable programming in C/C++ and want to modify DVM, this section briefly discusses the organization of the provided source code files.

```

; FILE: DvmSetup.ini                Copyright (c), Symmetric Research, 2008
;
; Example ini file for DVM using COM1 and showing Volts and Degrees.
;
; See DvmSetupExample.ini for an explanation of the ini syntax and
; a list of valid options.

; General ini parameters:

SerialPortName      = COM1
OutputFileName      = "DvmOutput.asc"
OutputFileComment   = "Example ini showing volts and degrees"
OutputFileShowHeader = ON
OutputFileShowIndex = ON
OutputFileShowTimeSec = ON
OutputFileShowTimeYmd = ON
RunMode             = CONTINUOUS

; Display 0 format and calibration parameters:

DisplayTitle 0 = "Voltage"
DisplayUnits 0 = "Volts"
DisplayPlaces 0 = 9
DisplayDigits 0 = 5
DisplaySlope 0 = 2.98511e-007
DisplayOffset 0 = -0.00119882

; Display 1 format and calibration parameters:
; Angular position with full range calibrated from 0 to 360.

DisplayTitle 1 = "Angle"
DisplayUnits 1 = "Degrees"
DisplayPlaces 1 = 6
DisplayDigits 1 = 2
DisplaySlope 1 = 2.16764e-005
DisplayOffset 1 = -0.0536708

```

Figure 3.2: Sample DVM .ini initialization file

The files `DvmLib.c` and `DvmLib.h` contain the major portion of DVM, the part that is shared between both the GUI and CMD versions. The files that start with `DvmGui` are used for the GUI version and the file `DvmCmd.c` is used for the CMD version. Both versions depend on the variables and functions defined in the `DvmLib` files.

Both `DvmGui` and `DvmCmd` can be built from the command prompt using the provided Makefile. It accesses various settings and macros contained in the `Compiler.mak` file in the `/SR/SER1CH-UA/Lib` directory and assumes that a batch file like `vsvars.bat` has been run to set the compiler environment. The `.sln` and `.vcproj` files can be used to build `DvmGui.exe` and `DvmCmd.exe` from

DVM .ini Keywords			
Keyword	Description	Num	Values
SerialPortName	Name of serial port to use	NO	COM1, COM2, COM3, COM4
OutputFileName	Name of file for saved data	NO	File name in quotes like "DvmOutput.asc"
OutputFileComment	Arbitrary text to help identify the saved data	NO	Descriptive comment in quotes like "Test 4"
OutputFileShowHeader	Switch controlling whether output file includes headers	NO	ON, OFF
OutputFileShowIndex	Switch controlling whether output file includes sample num	NO	ON, OFF
OutputFileShowTimeSec	Switch controlling whether output file includes time as seconds since 1970	NO	ON, OFF
OutputFileShowTimeYmd	Switch controlling whether output file includes time as year/month/day hour:minutes:seconds	NO	ON, OFF
RunMode	Whether to collect 1 or many data samples	NO	SINGLE, CONTINUOUS
DisplayTitle	Title for display N	YES	String in quotes like "Voltage"
DisplayUnits	Units for display N	YES	String in quotes like "Volts"
DisplayPlaces	Number of digit places for display N	YES	Integer like 9
DisplayDigits	Number of digits after the decimal for display N	YES	Integer like 5
DisplaySlope	Calibration slope value for display N	YES	Floating point number like 7.27444e-007
DisplayOffset	Calibration offset value for display N	YES	Floating point number like 0.0000953873

Figure 3.3: DVM .ini keyword table

inside the Visual Studio 2005 IDE.

Please refer to the comments inside the source code files themselves for more details.

3.5 CAL: A calibration program

To get the most accuracy from the DVM program described in section 3.1 on page 11 it is important to calibrate the A/D. The CAL program makes this calibration easy. Like DVM, there are two versions of CAL. CalCmd.exe is a console program that runs from the command line while CalGui.exe is a Windows Graphical User Interface (GUI) program that uses the Microsoft Foundation Classes (MFC) libraries.

Header:

General File Info:

```
HeaderFormatRev      = 2
DvmRev               = 2.01
FileName              = "DvmOutput.asc"
FileCreationDate     = 2008/02/01 18:57:12
IniFileName           = "DvmSetup.ini"
```

Ini Parameters:

```
SerialPortName       = COM1
OutputFileName        = "DvmOutput.asc"
OutputFileComment    = "Example ini showing volts and degrees"
OutputFileShowHeader = ON
OutputFileShowIndex  = ON
OutputFileShowTimeSec = ON
OutputFileShowTimeYmd = ON
RunMode               = CONTINUOUS
```

```
DisplayTitle 0 = "Voltage"
DisplayUnits  0 = "Volts"
DisplayPlaces 0 = 9
DisplayDigits 0 = 5
DisplaySlope  0 = 2.98511e-007
DisplayOffset 0 = -0.00119882
```

```
DisplayTitle 1 = "Angle"
DisplayUnits  1 = "Degrees"
DisplayPlaces 1 = 6
DisplayDigits 1 = 2
DisplaySlope  1 = 2.16764e-005
DisplayOffset 1 = -0.0536708
```

Data:

Sample	Volts	Degrees	Time (Sec)	Time (YMD HMS)
1	-0.00021	+0.02	1201892232.522	2008/02/01 18:57:12.522
2	+1.23895	+90.00	1201892233.684	2008/02/01 18:57:13.684
3	+2.47836	+180.00	1201892234.845	2008/02/01 18:57:14.845
4	+3.26040	+236.79	1201892236.007	2008/02/01 18:57:16.007
5	+4.95701	+359.99	1201892237.169	2008/02/01 18:57:17.169
6	+2.06543	+150.01	1201892238.330	2008/02/01 18:57:18.330
7	+0.30847	+22.43	1201892239.492	2008/02/01 18:57:19.492

Figure 3.4: Sample DvmOutput.asc output file in verbose mode.

Both versions of CAL read a preliminary set of control settings from a DVM initialization file such as DvmSetup.ini in the Calibrate directory and output the calibrated version of these settings into DvmSetupCal.ini. Don't forget to copy DvmSetupCal.ini to the DVM directory and rename it before running DVM again!

Running either version of CAL is simple. From Windows Explorer, double click on the RunCalCmd

-0.00021	+0.02
+1.23895	+90.00
+2.47836	+180.00
+3.26040	+236.79
+4.95701	+359.99
+2.06543	+150.01
+0.30847	+22.43

Figure 3.5: Sample DvmOutputBare.asc output file in bare mode.

or RunCalGui shortcut to start CalCmd.exe or CalGui.exe respectively. The RunCalCmd shortcut first starts a command prompt and then runs CalCmd.exe. You may also type in at a command prompt either program name followed by the name of the .ini file you wish to use. If no .ini file is specified, CAL will output a .ini file named DvmSetupDefault.ini containing the default settings. You should copy this file to a new name, modify it to have the settings you prefer, and run CAL again with the new .ini file name as a command line or shortcut argument.

Both versions of CAL require that certain information is entered and that two known input signals are presented to the SER1CH-UA for calibration measurements to be taken. The required information includes the display title (eg voltage) and name of the units to be used (eg volts), the value of one known input signal, and the value of a different known input signal. Because DVM displays the acquired data in two ways (for example in volts and in inches), appropriate calibration information is needed for both displays.

Once CalCmd.exe starts, it displays a prompt showing the serial port it will use. If this is not correct, please quit, edit the .ini file and try again. Also, see section 3.7 on page 19 for a brief discussion about how to choose a valid serial port.

Once the serial port selection is correct and you are ready to continue, CalCmd prompts you to enter the required calibration information starting with the display titles and units. Then apply a known input signal to the SER1CH-UA, enter its value and press a key to measure it. Repeat this for a second known input signal. CalCmd then computes the calibration results and outputs them to the DvmSetupCal.ini file.

Once CalGui.exe starts, it displays a dialog box showing the default serial port specified in the .ini file. If this is not correct, use the serial port dropdown list box to select the serial port that *is* connected to the SER1CH-UA.

CalGui provides you with text boxes for entering the required calibration information like the display titles and units. To perform the actual calibration, apply a known input signal to the SER1CH-UA, fill in the Value A text field for each display and click the MeasureA button to measure the A/D counts for that signal. Repeat with a second known input signal, Value B and the MeasureB button. The calibration results are updated each time a Measure button is pressed. Click the Save button to output the calibration results to the DvmSetupCal.ini file.

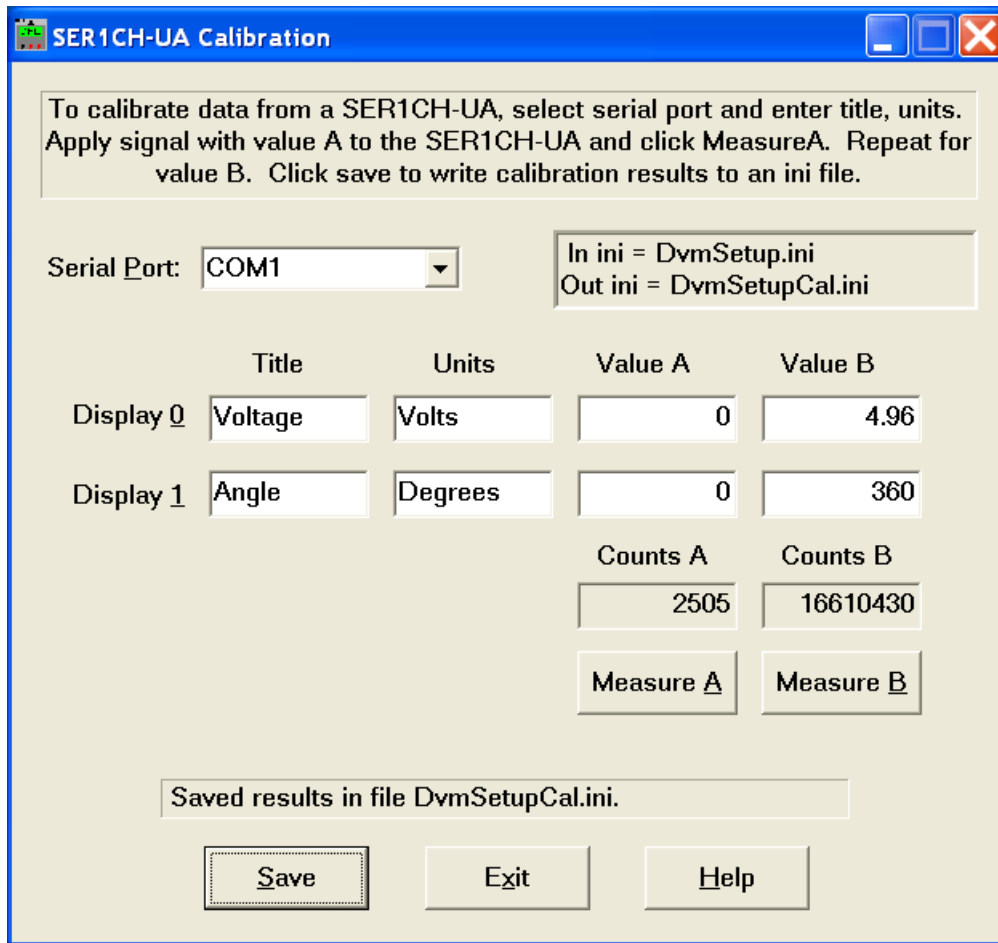


Figure 3.6: Screen shot of GUI version of CAL program

3.6 Modifying CAL

For those who are comfortable programming in C/C++ and want to modify CAL, this section briefly discusses the organization of the provided source code files.

The files CalLib.c and CalLib.h as well as DvmLib.c and DvmLib.h make up the part of CAL that is shared by both the GUI and CMD versions. The files that start with CalGui are used for the GUI version and the file CalCmd.c is used for the CMD version. Both versions depend on the variables and functions defined in the CalLib and DvmLib files.

Both CalGui and CalCmd can be built from the command prompt using the provided Makefile. It accesses various settings and macros contained in the Compiler.mak file in the /SR/SER1CH-UA/Lib directory and assumes that a batch file like vsvars.bat has been run to set the compiler environment. The .sln and .vcproj files can be used to build CalGui.exe and CalCmd.exe from inside the Visual Studio 2005 IDE.

Please refer to the comments inside the source code files themselves for more details.

3.7 Determining the correct serial port

For DVM and CAL to run correctly, it is important to select the serial port which is actually connected to the SER1CH-UA. The Windows Device Manager can help narrow down the choices by showing which ports are even available.

The Device Manager can be accessed graphically from the Control Panel or from MyComputer properties. It can also be accessed from a command prompt by typing devmgmt.msc and pressing ENTER.

Once the Device Manager window is showing, open or expand the Ports (COM & LPT1) class. The available serial ports are listed as Communications Port (COMx). Some additional COM ports may be assigned to other devices such as modems, but these will not be listed in the Ports class and can not be used with the SER1CH-UA.

Now that the possible serial ports have been identified, you can run DvmGui to test them out. Using the provided alligator clip cables, connect VREF from the red outer banana jack to VIN on the red middle banana jack. This provides a known voltage of about 4.96v to the A/D inputs.

Select one of the possible serial ports in the DvmGui dropdown list and click the run button. If you see the sample number incrementing and the voltage display showing 4.96 volts you have the right serial port. If not, click the stop button and repeat with the next serial port you want to try.

3.8 SER1CH DLL Library

Besides finished applications like DVM and CAL, the SER1CH-UA software also includes support for people writing their own programs by providing a function library for controlling the SER1CH-UA hardware. These functions can be statically linked to C programs and are also available as a Dynamic Link Library (DLL) under Windows or as a shared library (.so) under Linux.

The Ser1ch function library is written in C, but is designed to be called from other languages such as Visual Basic and LabView as long as C calling conventions are followed. The exact method for making function calls into a DLL library is specific to each language environment and typically requires declarations where the parameters being passed are described. Check your language documentation for details.

The sequence of calls to use the Ser1ch library functions is simple: First call Ser1chOpen to open and initialize the COM serial port and A/D. Then call Ser1chGetData each time you want a new A/D sample, followed by Ser1chScaleData if you want to scale the A/D counts to volts or other units. Finally, when you are finished acquiring data, call Ser1chClose to close the COM serial port so it is free for other programs to use.

Be sure to include the header file Ser1ch.h in any C source code using the Ser1ch library. This file contains function prototypes for ensuring the correct parameters are passed and defined constants that can help make your program more readable. When using dynamic linking, make sure the Ser1ch.dll library is on your Windows execution path or the libser1ch.so library is on your Linux

LD_LIBRARY path so it can be found at runtime.

The following table summarizes the Ser1ch library functions and their parameters. Each function is also described in more detail in the following subsections, along with a code fragment to show how they might be called from a C program.

In addition to the standard data types like integer and pointer to integer, the table also lists DEVHANDLE. This is special data type which is defined in SrHelper.h and varies depending on OS. Under Windows it is a (void *) and is equivalent to the Windows HANDLE type. Under Linux it is an integer. In both cases, the size is 32 bits and the parameter refers to a file handle which gives the OS a shorthand way to locate the information it needs for accessing the desired serial port.

Ser1ch Library Functions			
Function/Parm	IO	Type (#bits)	Description
Ser1chOpen			
return	O	DEVHANDLE (32)	Open the serial port and initialize A/D = BAD_DEVHANDLE on failure
SerialPortNumber	I	integer (32)	Serial port number 1=COM1, 2=COM2, ...
Error	O	pointer to integer (32)	Detailed error code
Ser1chGetData			
return	O	integer (32)	Read a data point and the PC time 1 for success, 0 for failure
Ser1chHandle	I	DEVHANDLE (32)	Handle to SER1CH-UA serial port
DataCounts	O	pointer to integer (32)	Data point as raw A/D counts
TimeSecSince1970	O	pointer to double (64)	PC time of data point as seconds since 1970
Error	O	pointer to integer (32)	Detailed error code
Ser1chScaleData			
return	O	integer (32)	Scale data point counts to other units 1 for success, 0 for failure
DataCounts	I	pointer to integer (32)	Data point as raw A/D counts
Slope	I	floating point double (64)	Slope of scaling formula
Offset	I	floating point double (64)	Offset of scaling formula
ScaledData	O	pointer to double (64)	Data point scaled to other units
Error	O	pointer to integer (32)	Detailed error code
Ser1chErrorMsg			
return	O	pointer to char (8)	Get string associated with error code Null terminated C-style string
Error	I	integer (32)	Detailed error code
Ser1chClose			
return	O	integer (32)	Close the serial port and A/D 1 for success, 0 for failure
Ser1chHandle	I	DEVHANDLE (32)	Handle to SER1CH-UA serial port
Error	O	pointer to integer (32)	Detailed error code

Figure 3.7: Ser1ch library function table

3.8.1 Open device

C usage:

```
#include "Ser1ch.h"

DEVHANDLE Ser1chOpen( int SerialPortNumber, int *Error );
```

The Ser1chOpen function opens and initializes the specified COM serial port and SER1CH-UA A/D. It should be the first library function called.

The first argument passed to Ser1chOpen is an integer from 1 to 9 that specifies the serial port to use. A value of 1 corresponds to COM1 under Windows or /dev/ttyS0 under Linux. 2 selects COM2, etc. You may want to check the Device Manager under Windows to see which COM ports are available on your machine as described in section 3.7 on page 19.

The second argument is a pointer to an integer that will be filled with an error code giving some detail about the failure if one occurs. An error code of 0 means the open succeeded. This argument is optional and will be ignored and not filled in if you pass Error = NULL. See Ser1ch.h for a list of the possible error codes. Or, use the Ser1chErrorMsg library function to get the message string associated with the integer error number.

Ser1chOpen returns a file handle to the open serial port in the form of a specially defined data type called a DEVHANDLE. Under Windows this is a (void *) and is equivalent to the Windows HANDLE type. Under Linux it is an integer. In both cases, the size is 32 bits and this file handle provides a shorthand way for the OS to access the selected serial port. If Ser1chOpen fails, the return value is BAD.DEVHANDLE which is defined as -1.

NOTE: Under Linux, Ser1chOpen will fail if you don't have read permission on the selected serial port. Type the command `ls -l /dev/ttyS0` to see the current permissions on COM1. The two easiest ways to add read permission for yourself are to join the group such as `uucp` or `nut` that owns the COM port or to type the command `chmod o+r /dev/ttyS0` as root to add world read permission to the COM port.

3.8.2 Read data

C usage:

```
#include "Ser1ch.h"

int Ser1chGetData( DEVHANDLE Ser1chHandle,
                  long *DataCounts,
                  double *TimeSecSince1970,
                  int *Error );
```

Once you have successfully called Ser1chOpen to open and initialize the serial port, you can call Ser1chGetData to read in a data point as raw A/D counts.

Ser1chGetData returns 1 for success and 0 for failure. It also takes an optional Error argument for returning more detailed information when a failure occurs. See Ser1ch.h for a list of the possible error codes, or use the Ser1chErrorMsg library function to get the message string associated with the error number.

The first argument to Ser1chGetData is the DEVHANDLE returned by Ser1chOpen containing a file handle to the open serial port. Ser1chGetData uses this handle when deciding which serial port to read.

The DataCounts argument to Ser1chGetData should be a pointer to a 32 bit long integer. The integer will be filled in with the A/D count value. The 24 bit A/D count is stored as a 32 bit integer with the top 8 bits always 0.

The optional TimeSecSince1970 argument returns the time when the A/D conversion occurred. Pass a NULL pointer if you wish to ignore this parameter. When used, the PC clock time as seconds since 1970 of when the A/D conversion occurred will be returned. This value can be converted into the more common year/month/day hour:minute:second format with the SrSecTimeSplit function from SrHelper.c or with the standard C runtime function gmtime.

3.8.3 Scale data

C usage:

```
#include "Ser1ch.h"

int Ser1chScaleData( long DataCounts,
                    double Slope,
                    double Offset,
                    double *DataScaled,
                    int *Error );
```

Once you have successfully called Ser1chGetData to read in a data point as raw A/D counts, you can call Ser1chScaleData to convert the counts to other more convenient units such as volts or inches.

Ser1chScaleData returns 1 for success and 0 for failure. It also takes an optional Error argument for returning more detailed information when a failure occurs. See Ser1ch.h for a list of the possible error codes, or use the Ser1chErrorMsg library function to get the message string associated with the error number.

DataCounts, the first argument to Ser1chScaleData, is a 32 bit integer containing the raw A/D counts returned by Ser1chGetData. The 24 bit A/D count is stored as a 32 bit integer with the top 8 bits always 0.

The second and third arguments, Slope and Offset, are 8 byte floating point doubles that are used to linearly convert from raw A/D counts to some other units such as volts. The scaled result is returned in the 8 byte floating point double parameter pointed to by DataScaled.

The conversion from A/D counts to converted scaled data is done using the slope and offset formula for a straight line:

$$DataScaled = slope * DataCounts + offset \quad (3.1)$$

It is easy to compute the slope and offset from the information generated by the CAL programs. Use CAL to measure the A/D counts for two known values of A/D input. Often the input will be specified in terms of volts, but it can be in other units such as degrees or inches too. This results in two (DataCounts,DataScaled) pairs, $(C1, S1)$ and $(C2, S2)$. These two calibration points for the straight line equation (3.1) then imply that:

$$\begin{aligned} slope &= (S2 - S1)/(C2 - C1) \\ offset &= (S1 - slope * C1) \end{aligned}$$

The CAL programs do this computation and save the resulting slope and offset values in the generated .ini file DvmSetupCal.ini. Use these values as inputs to the Ser1chScaleData function. Note that systematic offsets, such as potentiometer offsets and op amp V_{io} errors, can be removed by

specifying $S1 = 0$ when calibrating. Software offset correction is a powerful alternative to hardware trimming.

For the SER1CH-UA, the A/D counts always range from 0x00000000 to 0x00FFFFFF, (0 to 16,777,215 decimal), and the input voltages at which these count values occur are approximately 0 and 4.96 volts. Using these values, the slope in volts/count should always be approximately $4.96/16777215 = 2.96e-7$. The offset should always be approximately 0 because zero input volts maps approximately into 0 counts. Even if you don't have exact calibration numbers for a specific SER1CH-UA, you can still get reasonably good results for voltage calibration by using the approximate slope and offset numbers:

$$\begin{aligned} \text{slope} &\approx 2.96e-7 \text{ volts/count} \\ \text{offset} &\approx 0.0 \text{ volts} \end{aligned}$$

3.8.4 Get error message string

C usage:

```
#include "Ser1ch.h"

char *Ser1chErrorMsg( int Error );
```

The `Ser1chErrorMsg` function takes an error number as input and returns a pointer to a null terminated string containing the corresponding message text.

Also see the C include file `Ser1ch.h` and source file `Ser1ch.c` for a listing of the error numbers and corresponding strings.

3.8.5 Close device

C usage:

```
#include "Ser1ch.h"

int Ser1chClose( DEVHANDLE Ser1chHandle, int *Error );
```

The Ser1chClose function closes the COM serial port. It should be the last library function called. Ser1chClose returns 1 for success and 0 for failure.

The first argument to Ser1chClose is the DEVHANDLE returned from the Ser1chOpen function. This is a 32 bit value representing a file handle to the open serial port. Under Windows it is equivalent to the Windows HANDLE type.

The second argument is a pointer to an integer that will be filled with an error code giving more detail than the simple (0,1) function return value. This argument is optional and will be ignored if you pass a NULL pointer. See Ser1ch.h for a list of the possible error codes. Or, use the Ser1chErrorMsg library function to get the message string associated with the integer error number.

3.8.6 Code example

The following C program is an example of calling the Ser1ch DLL function library. For simplicity, sometimes only the function success or failure is processed and not the full error code.

```
#include <stdio.h>
#include "Ser1ch.h"

#define SLOPE      2.96e-7F    // < approximate SER1CH-UA volts/count slope
#define OFFSET    0.0F        // < approximate SER1CH-UA offset in volts
#define FMT_STRING "Data pt %02d is 0x%08lX counts or %lf volts at Time %lf\n"

int main( void ) {

    int i, Success, Error; long Data; double Scaled, Time;
    DEVHANDLE Ser1chHandle;

    // Open the SER1CH on COM1.

    Ser1chHandle = Ser1chOpen( 1, &Error );

    if ( Ser1chHandle == BAD_DEVHANDLE ) {
        printf( "Failed to open the SER1CH-UA on COM1 ...\n");
        return( 1 );
    }

    // Read and scale 50 data points.

    for ( i = 0 ; i < 50 ; i++ ) {

        Success = Ser1chGetData( Ser1chHandle, &Data, &Time, &Error );

        if ( Success == 1 ) {
            Ser1chScaleData( Data, SLOPE, OFFSET, &Scaled, NULL );
            printf( FMT_STRING, i, Data, Scaled, Time );
        }

        else
            printf("Data pt %02d failed because %s\n", i, Ser1chErrorMsg( Error ) );

    }

    // Close.

    Success = Ser1chClose( Ser1chHandle, &Error );

    return( 0 );
}
```

Figure 3.8: Sample program calling Ser1ch library functions

Chapter 4

Sigma Delta A/D converters

UNDER CONSTRUCTION ...

Review that sigma delta A/D converters achieve their resolution by oversampling and averaging down and the boxcar averages result in a sinc response.

Chapter 5

60Hz and 50Hz power line rejection

Because of their internal oversampling and digital filtering, sigma delta converters such as the LTC2400 used in the SER1CH-UA are capable of excellent power line noise rejection.

The internal sampling rate of the SER1CH-UA LTC2400 is automatically set to place the first notch of the sigma delta digital filter at 60Hz. International users may change the oversampling rate to place the first notch at 50Hz.

UNDER CONSTRUCTION ...

Review that the power line noise rejection is typically greater than 120 dB.

Chapter 6

Sampling rate

Currently the sampling is set at 1Hz by a timer tick running on the PC.

UNDER CONSTRUCTION ...

Review the LTC2400 maximum sampling rate of 7Hz.

Chapter 7

Resolution

The SER1CH-UA uses a 24 bit A/D converter. Plot the noise floor versus time and compare with the LTC2400 spec sheet.

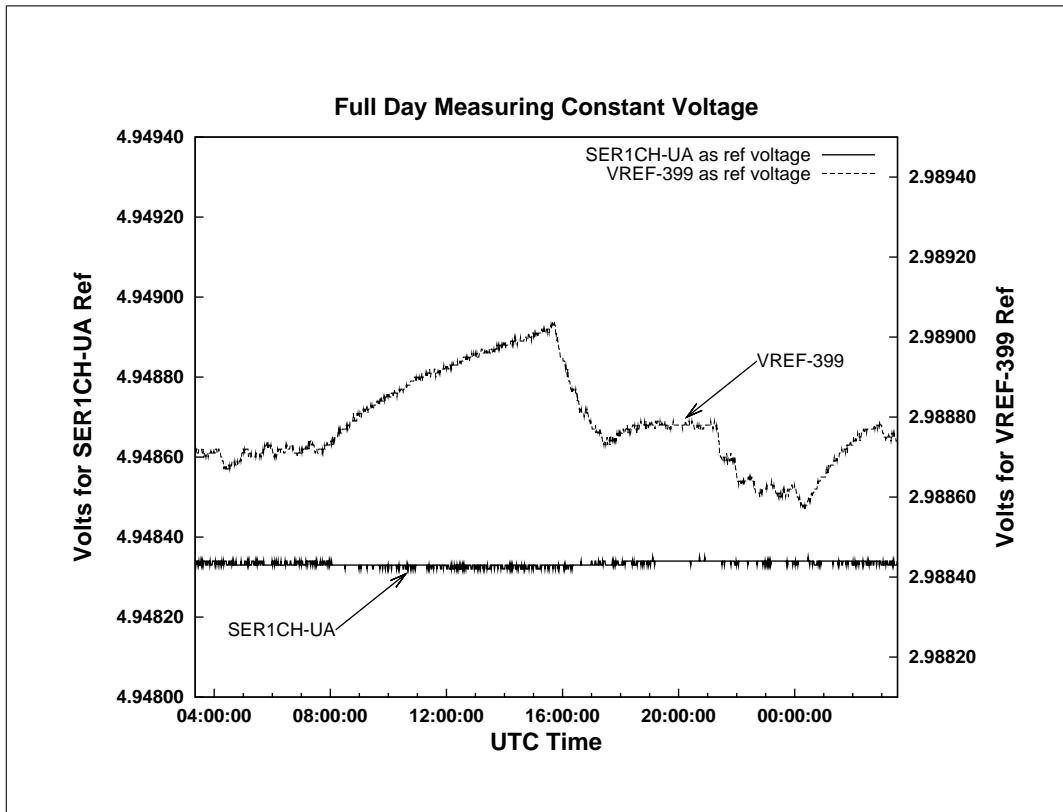


Figure 7.1: SER1CH-UA long term resolution plot

More descriptive text about the resolution plot ... plus more plots over different time scales ...

For ways to battle back against thermal variation, see the *Ratiometric measurements* chapter, or measure the local temperature with a temperature gauge, or just keep the temp of the SER1CH-UA constant ...

Chapter 8

Ratiometric measurements

Electronic components vary with changes in temperature and those variations can make repeatable precision A/D measurements difficult. Although all electronic components have thermal variation, one of the main contributors to thermal error in any A/D system is the *voltage reference* used by the A/D converter itself. Ratiometric measurements are one easy and effective way to reduce such errors. Let's take a look at some of the basic equations for an A/D converter to see how ratiometric methods work.

A/D converters accept an analog input voltage and produce a digital output count. The converter determines the output by comparing the input against a reference voltage. Denoting the A/D reference voltage by V_{ref} one might write:

$$AD_{counts} = (AD_{input} / V_{ref}) \cdot AD_{FullScaleCounts} \quad (8.1)$$

where the $AD_{FullScaleCounts}$ is fixed by the architecture of the converter and the number of bits it returns. In the case of a 24 bit converter, the full scale counts might be 0xFFFFF, and an input voltage that is half of the reference voltage would return 0x7FFFF as output.

For most systems, the reference voltage is established by a component separate from the A/D converter and any change in the value of V_{ref} will result in a change in the AD_{counts} . For the SER1CH-UA the reference voltage is established by a 2.5 volt LM4040A bandgap device. This low power, low noise part has a typical 15 ppm/°C thermal variation, and so the reference may vary by $(15 \cdot 10^{-6}) \cdot 2.5 \text{ volts} = 37.5 \text{ microvolts}/^\circ\text{C}$.

Now, a wide range of *passive sensors* have an output voltage proportional to their setting *and to their excitation voltage*:

$$AD_{input} = Sensor_{output} = Sensor_{setting} \cdot Sensor_{excite} \quad (8.2)$$

One of the many classic sensors of this type is a potentiometer. If we arrange the measurement so $Sensor_{excite} = V_{ref}$, that is *the A/D reference is also used to excite the sensor*, then V_{ref} will cancel from the top and bottom of equation (8.1) and the final A/D counts becomes:

$$AD_{counts} = Sensor_{setting} \cdot AD_{FullScaleCounts} \quad (8.3)$$

When used in this way, the A/D output counts are independent of the exact value of the reference voltage and the system is insensitive to reference thermal variations! This is referred to as a *ratiometric* measurement.

As a specific example, let's work through the equations using a potentiometer. Many applications use potentiometers as position or angular sensors. The circuit for a potentiometer hooked up to the SER1CH-UA in ratiometric fashion is:

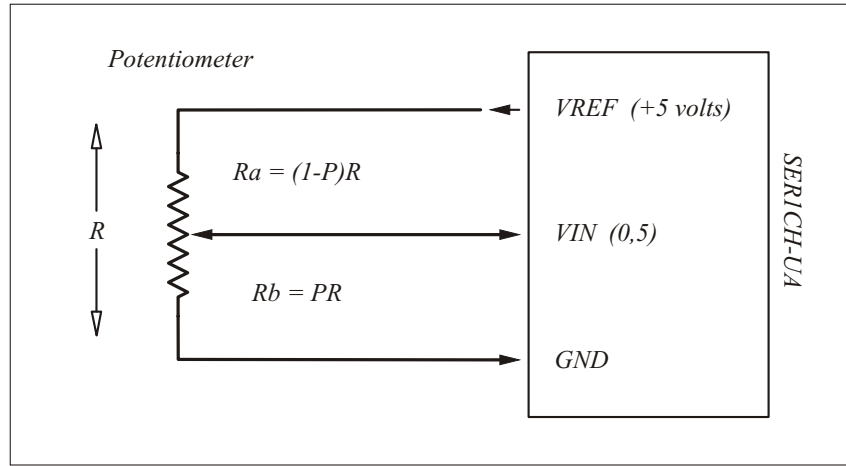


Figure 8.1: Potentiometer parameters

The connections to the SER1CH-UA are simple. The three terminals of the potentiometer are connected directly to the three banana jacks of the SER1CH-UA. No conditioning circuitry is required, and VREF provides the excitation voltage for the potentiometer.

In this figure, R is the total end to end resistance of the potentiometer, while R_a and R_b are the resistances above and below the wiper. P denotes the position of the wiper and varies from 0 to 1. Assuming no load on the potentiometer output, the wiper voltage is given by the resistor divider equation:

$$Potentiometer_{output} = (R_b / (R_a + R_b)) \cdot V_{ref} = (R_b / R) \cdot V_{ref} \quad (8.4)$$

which in terms of wiper position, $R_b = P \cdot R$, is simply:

$$AD_{input} = Potentiometer_{output} = P \cdot V_{ref} \quad (8.5)$$

Substituting this into equation (8.1) results in:

$$AD_{counts} = P \cdot AD_{FullScaleCounts} \quad (8.6)$$

For a particular potentiometer position P , the same count value is returned regardless of the exact value of the A/D reference voltage.

In fact, this is really a bridge or differential measurement. The A/D forms one side of the bridge, while the potentiometer forms the other side. As long as the voltage at the top of each leg of the bridge is the same, all that matters is the the *ratio* of the resistors on each side. Hence the term ratiometric measurement.

On the SER1CH-UA, a buffered copy of the A/D reference voltage is provided on a banana jack for ratiometric applications. The VREF source is capable of supplying 10ma, so keep the potentiometer and other load resistance greater than 500 ohms to avoid drawing too much current. A 10K pot is often ideal. The SER1CH-UA analog input also has an input impedance of 10M ohms so simple passive sensors like potentiometers will not be loaded.

Just how good are ratiometric techniques?

The following graphs are the result of one experiment . . . compare a divider driven by the VREF-399 and a ratiometric SER1CH-UA.

For a hands on demonstration using a 10 turn potentiometer in ratiometric fashion see the *Example applications* chapter of this User Manual.

Chapter 9

Scaling active sensor voltages

The full scale input range of the SER1CH-UA is 0 to +5 volts. If you have an *active* sensor with a range other than this, you will probably want to scale and bias its output to be compatible with the SER1CH-UA (0,5) input.

Scaling and biasing is usually *not required for passive sensors*. Passive sensors such as potentiometers typically have their full scale output range set by their excitation voltage. If the excitation is supplied by the SER1CH-UA VREF, a full scale sensor range of (0,5) results naturally and no scaling or biasing will be required. Such techniques are covered in the *Ratiometric measurements* chapter and have the benefits of excellent TC temperature coefficient performance.

Active sensors on the other hand typically include transistors and op amps in their design. They require an independent power supply and have a fixed full scale output range. When working with these types of sensors, it is desirable to map their full scale range into the SER1CH-UA (0,5) so none of the 24 bit A/D span is lost.

A few of the more popular circuits for mapping active sensors are reviewed in this chapter. In most cases, precision low noise mapping can be done with a few resistors, while for applications requiring gain an op amp may be required.

For further information about scaling and biasing beyond this chapter refer to:

- the *Example applications* chapter of this User Manual for a (-10,+10) demo
- the SR application note, *Scaling and Biasing Analog Signals* for additional theory

SR application notes can be found on the web at: www.symres.com/download

9.1 Resistive scaling and biasing

Active sensors typically have output voltage ranges corresponding to the technology of the op amps they use. The most common ranges are (-5,+5) and (-10,+10) volts. (-5,+5) may result if the sensor uses low voltage rail to rail op amps, while (-10,+10) may result if the sensor uses traditional bipolar op amps. This section presents resistor circuits for mapping these two ranges into the SER1CH-UA (0,5) as well as the modifications required for other ranges.

The resistive circuits given here are desirable for use with active sensors. Besides being simple, they introduce very little noise. Generally, op amp buffering is not required for these circuits because active sensor outputs are designed to drive the moderate input impedance of these resistive circuits and the high input impedance of the SER1CH-UA presents no load to the divider output.

The first circuit in Figure 9.1 is suitable for mapping (-5,+5) volts into (0,5):

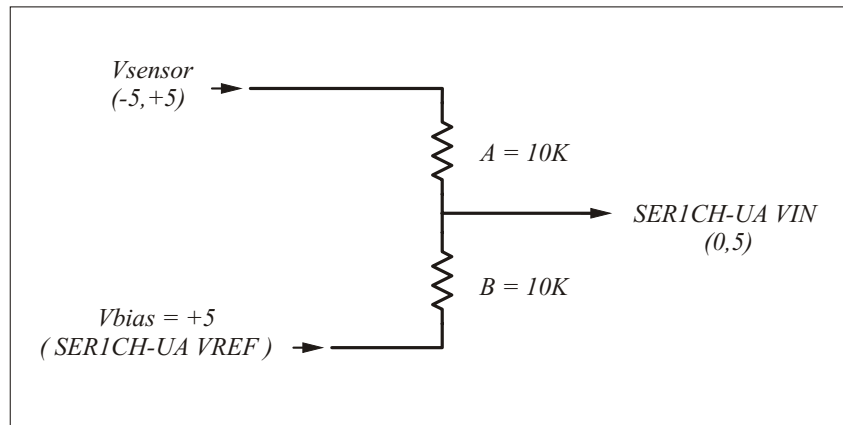


Figure 9.1: Two resistor scaling and biasing (-5,+5) into (0,5)

This (-5,+5) design uses two resistors and the SER1CH-UA VREF as its bias source. Use 1/4 watt metal film resistors with leads for easy construction. Carbon film resistors are not ideal because they have poor noise and TC performance and offer only a small cost savings. See the *Extra supplies* chapter for vendors carrying 1% metal film resistors. See the SR application note: *Scaling and Biasing Analog Signals*, for the theory behind how this circuit works.

When using this circuit, note that if the V_{sensor} input is allowed to float it will drift to +5 volts and *not* ground. If you measure the floating input with a Fluke or other voltmeter it will read +5 volts. The SER1CH-UA itself will also return +5 volts if the V_{sensor} input is allowed to float. To work correctly, the input must be driven by a source with a low impedance output such as an active sensor. If you want to temporarily hardwire the input low, connect it to ground with a wire.

The next circuit in Figure 9.2 is suitable for mapping (-10,+10) active sensors into (0,5). This design uses three resistors and the SER1CH-UA VREF as its bias source. As with the previous circuit, use 1/4 watt metal film resistors for best performance. If you don't have a 5K ohm resistor on hand, build one from two 10K resistors in parallel. For a hands on demo of this circuit, see the *Example applications* chapter.

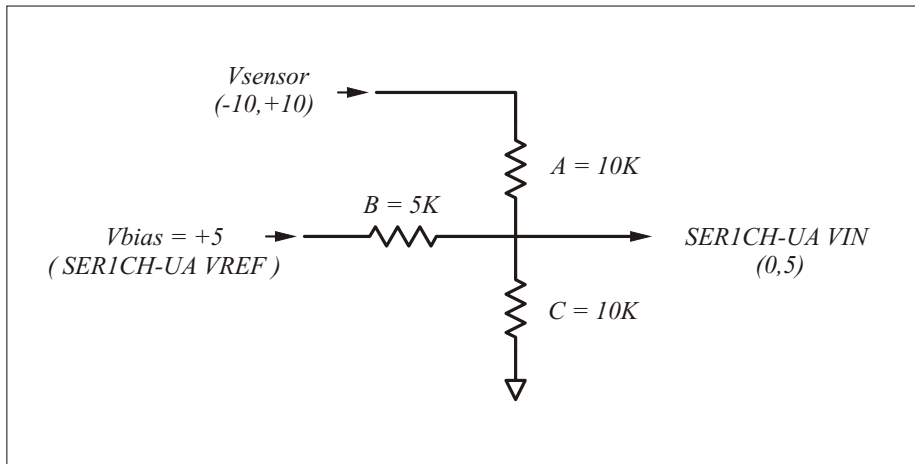


Figure 9.2: Three resistor scaling and biasing (-10,+10) into (0,5)

As with the two resistor circuit, if the input of the three resistor circuit is allowed to float, its output will drift to a non zero voltage. With the values in Figure 9.2, it will drift to 3.3 volts. Connect the V_{sensor} input to ground if you wish to temporarily drive the input low for a test.

For other input ranges, the circuit in Figure 9.2 can be modified with different resistor values and still use the SERICH-UA VREF as the bias source. The table in Figure 9.3 lists some possible values. The most common (-10,+10) case is marked with a * star.

Bipolar input	ΔV_{in}	V_{out}	A(Kohm)	B(Kohm)	C(Kohm)
(+/-) 05.0	10.0	(0,5)	10.000	10.000	*****
(+/-) 06.0	12.0	(0,5)	10.000	8.333	50.000
(+/-) 07.0	14.0	(0,5)	10.000	7.143	25.000
(+/-) 08.0	16.0	(0,5)	10.000	6.250	16.667
(+/-) 09.0	18.0	(0,5)	10.000	5.556	12.500
* (+/-) 10.0	20.0	(0,5)	10.000	5.000	10.000
(+/-) 11.0	22.0	(0,5)	10.000	4.545	8.333
(+/-) 12.0	24.0	(0,5)	10.000	4.167	7.143
(+/-) 13.0	26.0	(0,5)	10.000	3.846	6.250
(+/-) 14.0	28.0	(0,5)	10.000	3.571	5.556
(+/-) 15.0	30.0	(0,5)	10.000	3.333	5.000
(+/-) 16.0	32.0	(0,5)	10.000	3.125	4.545
(+/-) 17.0	34.0	(0,5)	10.000	2.941	4.167
(+/-) 18.0	36.0	(0,5)	10.000	2.778	3.846
(+/-) 19.0	38.0	(0,5)	10.000	2.632	3.571
(+/-) 20.0	40.0	(0,5)	10.000	2.500	3.333

Figure 9.3: Three resistor values for various input ranges with $V_{bias} = VREF$

9.2 Op Amp scaling and biasing

Adding op amps to the resistive circuits in the previous section provides further design possibilities for scaling and biasing. Besides increasing the input impedance of a resistive divider, op amps can perform signal conditioning such as adding gain or filtering. This section reviews two of the many possibilities.

The design in Figure 9.4 uses a unity gain buffer to increase the input impedance of its 10K/3.3K scaling and biasing pair into the Mohm range. The 3.3v bias required for this circuit is derived from the 10K/20K divider which is op amp buffered to be stable and constant.

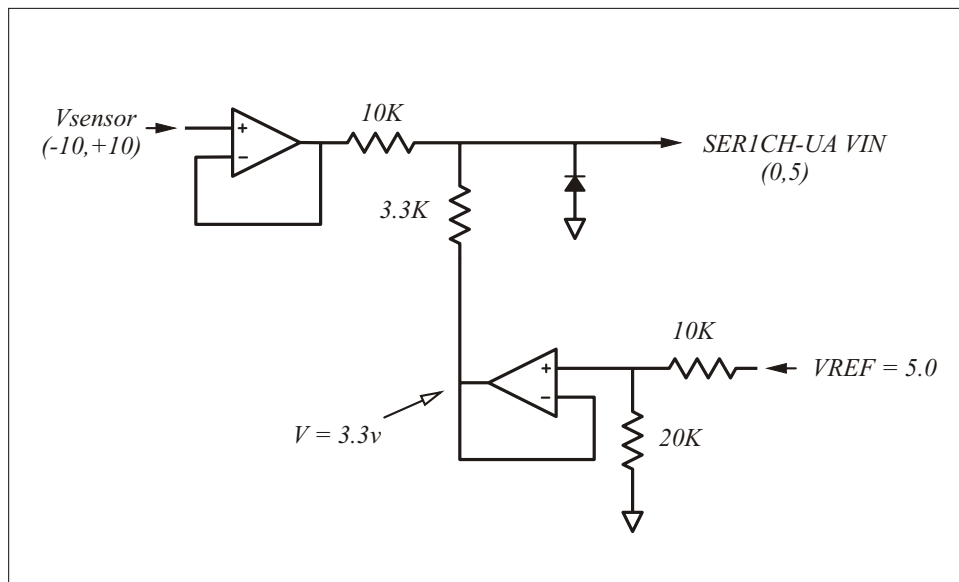


Figure 9.4: Op amp unity gain buffer with scaling

Note that the input amplifier in Figure 9.4 will need a split power supply in order for its non inverting + input to respond to negative signals. Typically a supply like +/-12 will be required. By comparison, the purely resistive circuits in the previous section only require the single positive 5v V_{REF} from the SER1CH-UA. We assume op amp users are familiar with the power supply requirements for these devices.

You may also wish to include a small signal diode as in Figure 9.4 to prevent accidental negative excursions from being presented to the SER1CH-UA (0,5) input. This might happen if the $(-10, +10)$ input went more negative than -10 volts. No damage will occur to the SER1CH-UA with negative inputs, but preventing this from happening is a good precaution.

Inexpensive bipolar op amps such as the LM358 (dual) or LM324 (quad) can offer surprisingly good performance in such circuits. With their multiple op amps per IC package, such devices make it easy to build circuits requiring multiple amps and buffers. When using generic parts, remove amplifier offset errors in software.

The circuit in Figure 9.5 adds gain to the input signal. After amplifying by 10, the signal is scaled and biased into the SER1CH-UA (0,5) range with the 10K resistor pair. When building a design like this, it is ideal to place the gain amplifier out near the sensor. Low voltage levels like +/-500mV input are much more sensitive to noise pickup than high voltage levels. Any long cable could then be placed between the output of the gain amplifier and the 10K divider pair. Check the signals on the cable to make sure its capacitive loading has not made the gain amplifier start oscillating.

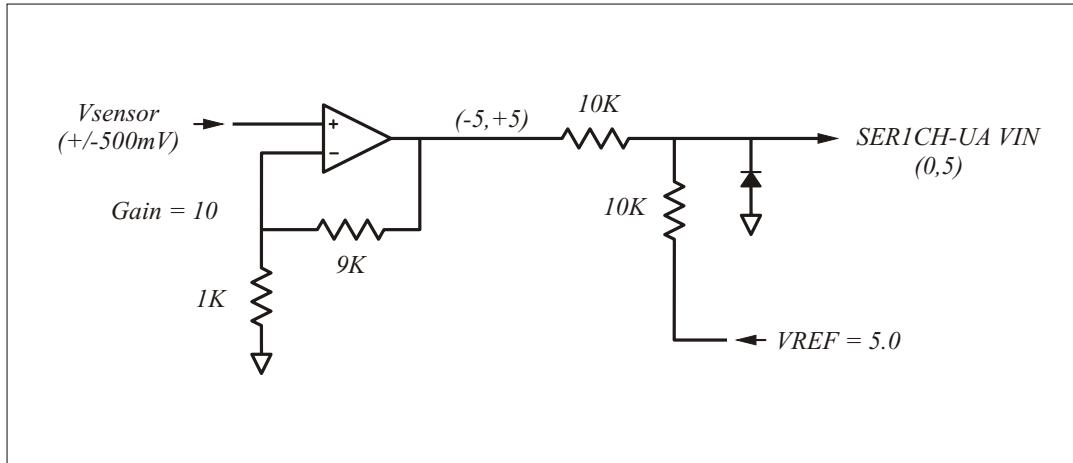


Figure 9.5: Op amp buffer with gain of 10

With the small input range of this circuit, it is easy for a signal more negative than -500mV to get hooked up to the input by accident. A small signal diode may be added as shown to clamp out of range negative voltages.

For single ended circuit such as this, do not allow ground currents for other devices to flow on the ground wire of the cable connecting the op amp to the SER1CH-UA. Excessive common mode currents, such as those from motors, will bounce the op amp and sensor ground and be apparent in the SER1CH-UA measurements.

Chapter 10

Example applications

The following are a few hands on demonstrations of using the SER1CH-UA. Several of these examples show how to implement topics covered in previous chapters. These are only starting points and should be modified as necessary for the work at hand.

10.1 Basic voltage measurements with probes

Measuring a voltage with the SER1CH-UA is similar to using other voltmeters, with the voltage source being connected between the red VIN and black GND banana jacks. Figure 10.1 shows a pair of AA batteries in series, and the alligator test leads supplied with the system used to measure their voltage.

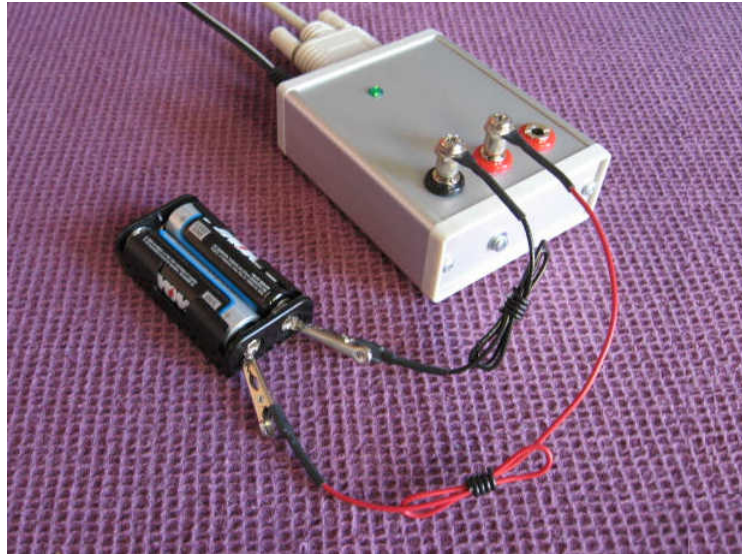
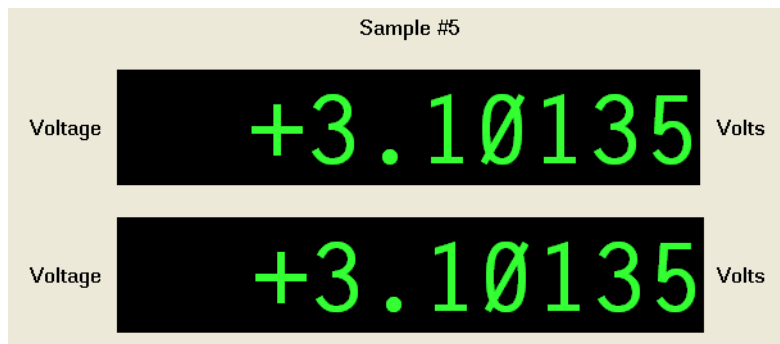


Figure 10.1: Two lead voltage measurement of a battery

*** WHAT HAPPENS IF THE BATTERY IS HOOKED UP IN REVERSE ??????

Since two AA cells in series is approximately 3 volts, if you use the DVM program with the startup file DvmVoltage.ini you should see the following on your computer screen:



In this particular DVM display, both readouts have been calibrated in volts. For many applications it is useful to calibrate the second DVM readout into other more physical units proportional to the

voltage, such as potentiometer position, temperature, etc. It is convenient to have two readouts so you can monitor the raw sensor voltage while also displaying the physically sensed quantity. See the *Software* chapter of this User Manual and the CAL program for more details.

Besides using the alligator leads supplied with the system, you can also use standard red/black banana plug test leads. In Figure 10.2, a pair of Pomona test leads are shown connected to the SER1CH-UA banana jacks. If you need a pair of heavy gauge test leads, they may be ordered as an accessory for the basic system.

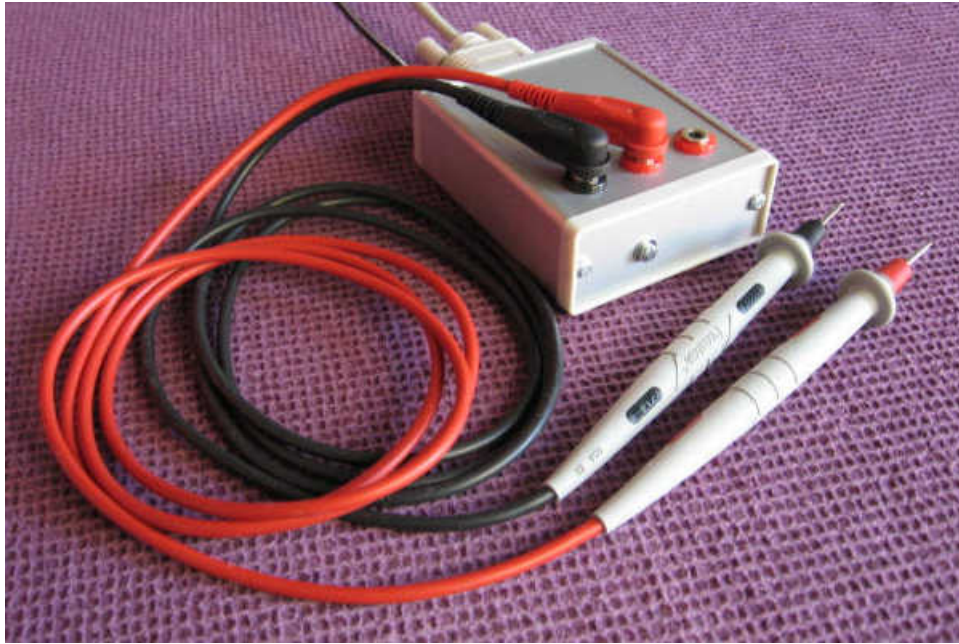


Figure 10.2: Pomona test leads for voltage measurements

When making two wire voltage measurements as in Figures 10.1 and ??, be aware the black GND banana jack is ultimately connected to the power ground and enclosure of the PC via the DB9 RS232 serial port connector. The analog inputs *are not floating* as they are with a hand held voltmeter. Also be aware current flowing in the ground system can cause noise in your measurements. Use a star layout for ground and ideally keep your sensor at one end of the star without system ground currents passing through its circuit.

Another difference with typical hand held voltmeters is the voltage between the VIN and GND jacks should be in the range of (0,5) volts. The system will tolerate moderate voltages outside of this range, but *do not attempt to measure 110 vac from the wall socket with this device*. Doing so will void the warranty. If you need to measure voltages outside the (0,5) range, please use the techniques in the *Scaling active sensor voltages* chapter in this User Manual, or the (-10,+10) section further on in these examples.

Simply being able to measure and record voltages provides many possibilities for interesting appli-

cations. As an example, Figure 10.3 shows using a solar cell as a light sensor. The voltage generated by the solar cell is a measure of the amount of light shining on it. The cell in Figure 10.3 has a voltage ranging from 0 volts at total darkness to approximately 3 volts at full brightness.

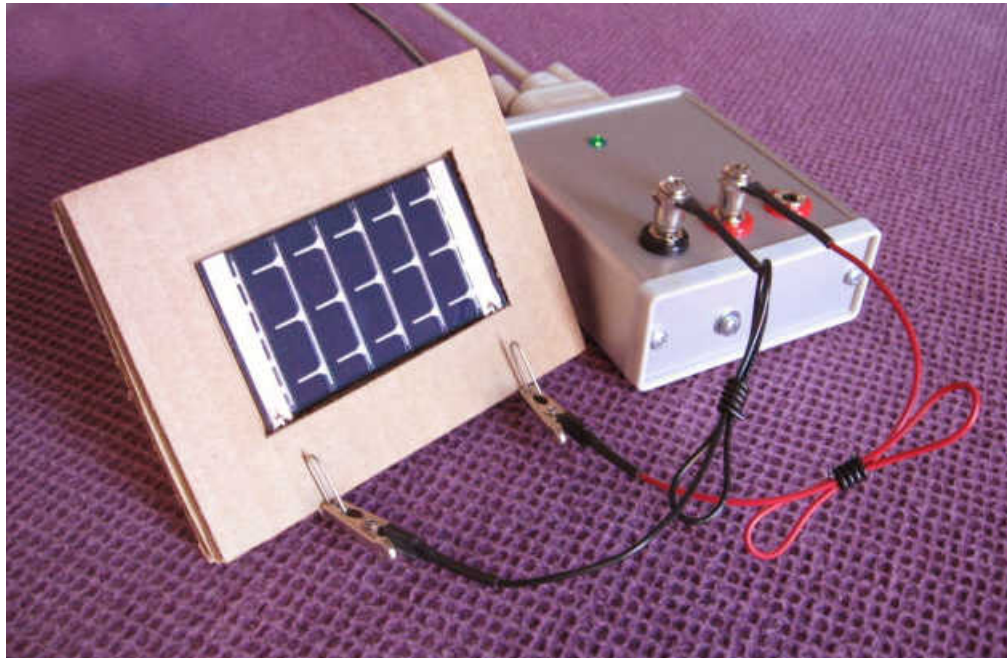


Figure 10.3: Using a solar cell as a light level meter

Not only can you measure the solar cell voltage with DVM, you can also calibrate the second DVM readout to display the value in light level units. Having a second readout calibrated into physical units makes it easy to determine the current condition of a sensor.

Besides its real time GUI readout display, DVM also saves its data values to a disk file. This makes it easy to review results after a long data run. The next chapter section shows how to take a DVM output file and plot the results with GnuPlot on the computer screen or printer.

10.2 Plotting results with GnuPlot

After acquiring data with the SER1CH-UA, you may want to plot the values. GnuPlot is a useful public domain program for displaying results on the computer screen or printer in a variety of formats. Copies can be downloaded for free from the web at:

www.sourceforge.net

or found by searching the web on the keywords GnuPlot. PDF documentation is included with GnuPlot when it is unpacked. Programs such as Xcel and MatLab may also provide useful plotting tools for users already having those programs.

The techniques for plotting results with GnuPlot or similar programs all follow the same pattern. First, run the SER1CH-UA DVM program to acquire values and save the results to an output disk file. For this example we assume data from the solar cell in the previous section has been saved in the file DvmOutputSolar.asc. The DVM output files are in ASCII format with the data in columns which can be imported into plotting programs. The .asc output files may also be reviewed in text editors such as Notepad. The following is a portion of the DVM .asc output data file that might result:

Data:

Sample	Volts	OneToTen	Time (Sec)	Time (YMD HMS)
1	+0.7	+1.42	1202171952.468	2008/02/05 00:39:12.468
2	+0.7	+1.43	1202171953.629	2008/02/05 00:39:13.629
3	+0.7	+1.43	1202171954.791	2008/02/05 00:39:14.791
4	+0.7	+1.45	1202171955.953	2008/02/05 00:39:15.953
5	+0.7	+1.49	1202171957.114	2008/02/05 00:39:17.114
6	+1.0	+2.01	1202171958.276	2008/02/05 00:39:18.276
7	+0.9	+1.90	1202171959.438	2008/02/05 00:39:19.438
8	+1.0	+1.99	1202171960.599	2008/02/05 00:39:20.599
9	+1.2	+2.34	1202171961.761	2008/02/05 00:39:21.761

Figure 10.4: DVM output file fragment from DvmOutputSolar.asc

To produce an output file with these data columns, DVM was run with the .ini startup parameters shown in Figure 10.5. Besides the general output format, this .ini file also specified the second display should be formatted in light level units of 1 to 10. You may wish to calibrate into other light level units as appropriate.

Once the DvmOutputSolar.asc file has been acquired, start GnuPlot and issue the commands in Figure 10.6 at its prompt.

You can also save the commands in Figure 10.6 to a file and then run them in GnuPlot with the load command. The file Solar.gp in the examples software directory has the complete set of commands

```

; General ini parameters:

OutputFileName      = "DvmOutputSolar.asc"
OutputFileComment  = "Measuring solar panel response"
OutputFileShowHeader = ON
OutputFileShowIndex = ON
OutputFileShowTimeSec = ON
OutputFileShowTimeYmd = ON
RunMode            = CONTINUOUS

; Display 1 format and calibration parameters:

DisplayTitle 1 = "Light Level"
DisplayUnits 1 = "OneToTen"
DisplayPlaces 1 = 6
DisplayDigits 1 = 2
DisplaySlope 1 = 6.02121e-007
DisplayOffset 1 = -0.00147941

```

Figure 10.5: DVM startup file fragment from DvmSetupSolar.ini

```

# SET FILE FORMAT PARAMETERS:

DataFile = "DvmOutputSolar.asc"
ColSample = 1
ColVolts = 2
Collight = 3
ColSeconds = 4
ColTime = 5

# SET AXIS DATA LIKE TIC MARK STYLE, FORMAT, LABEL AND RANGE:

set xtics nomirror font "Helvetica Bold,14"
set ytics nomirror font "Helvetica Bold,14"
set xlabel "Sample" font "Helvetica Bold,18"
set ylabel "Volts" font "Helvetica Bold,18"
set xrange [0:120] # About 2 minutes of data
set yrange [0:5] # Solar panel outputs ~3.3v in strong sunlight

# PLOT THE SOLAR DATA:

set title "SER1CH-UA Solar Panel Data" font "Helvetica Bold,18"

plot DataFile index 1 using ColSample:ColVolts linestyle 1 title "Solar panel voltage"

```

Figure 10.6: GnuPlot commands for plotting DvmOutputSolar.asc

required to produce the following graph. The core GnuPlot command is `plot`. With that single command you can display simple plots. Additional commands should be used for features like graph titles. Refer to the GnuPlot documentation for more information.

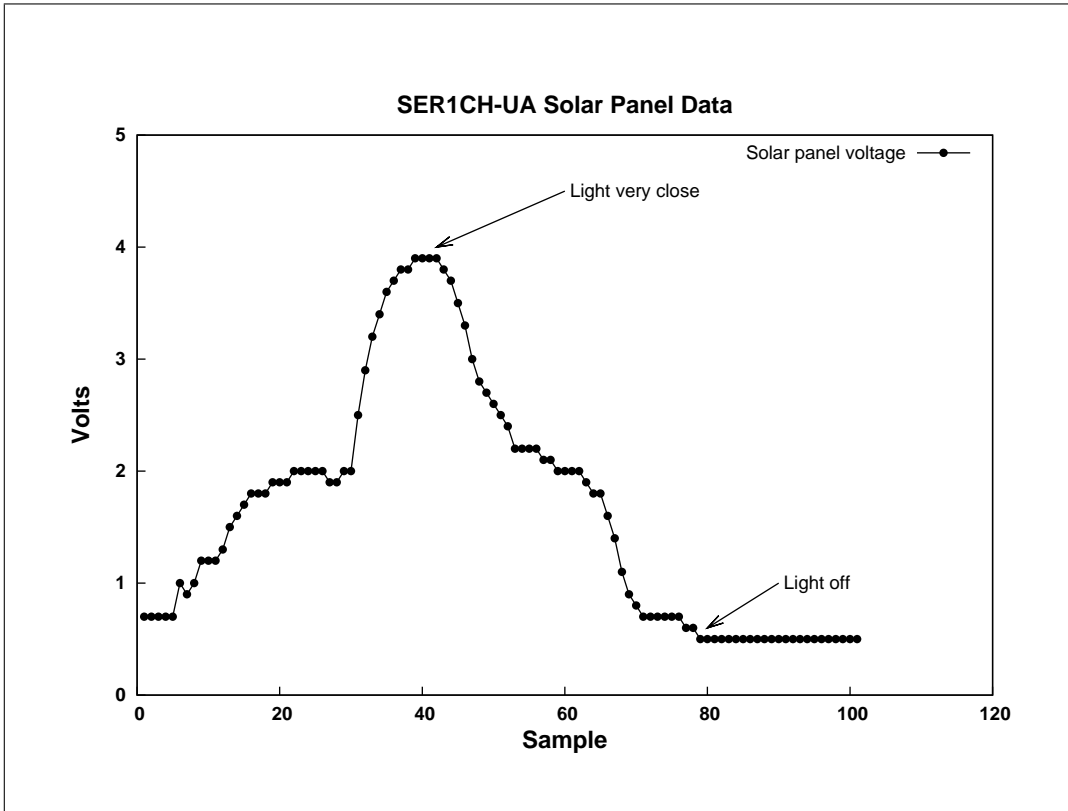


Figure 10.7: SER1CH-UA solar cell data example plot

10.3 Ratiometric potentiometer

Three wire ratiometric measurements are an excellent way to minimize errors due to temperature drift. The math behind such techniques is covered in the *Ratiometric measurements* chapter earlier in this User Manual. The demo reviewed here shows the connections for a three wire measurement using a precision 10 turn wire wound potentiometer.

For a ratiometric measurement, the A/D reference voltage provided on the red VREF banana jack should be used as the excitation voltage for the potentiometer, and the potentiometer wiper should be connected to the VIN jack. The photo in Figure 10.8 shows the connections. For a schematic, see the *Ratiometric measurements* chapter. Here, a 10K ohm potentiometer has been used. This means a total of $5V/10K = 0.5$ milliamps is flowing through the potentiometer. This is good, it is a large enough current to avoid noise problems, but still small enough to avoid putting a high load on the VREF output.

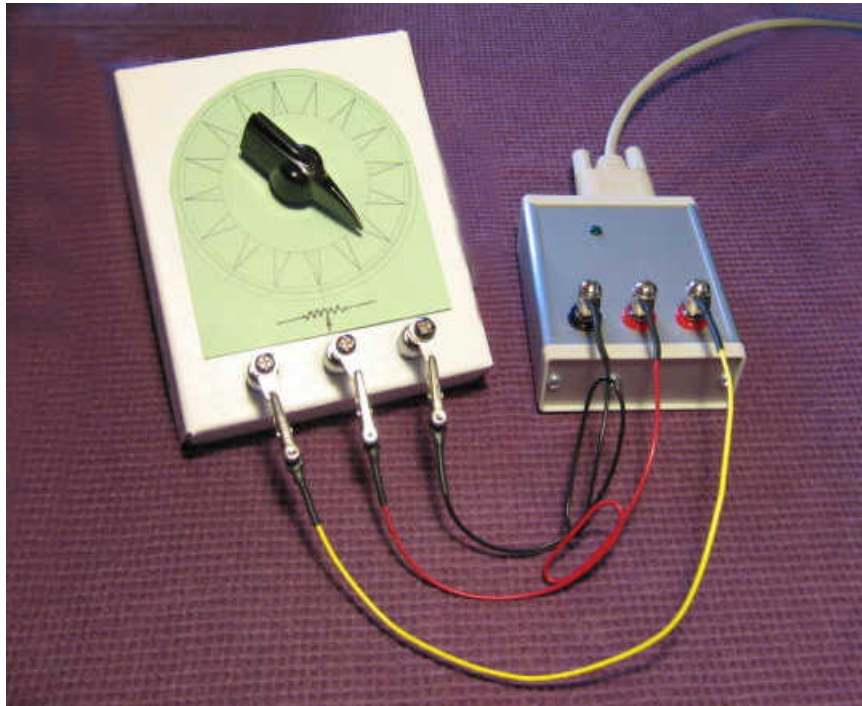


Figure 10.8: 10 turn potentiometer, SER1CH-UA connections

The connections on the bottom side of the potentiometer as in Figure 10.9 are simple wires to the three potentiometer terminals.

For an application of like this type you probably want the second DVM readout display to be calibrated in Turns. See the .ini startup file `Examples/DvmPotentiometer.ini` for an example. Generating suitable calibration parameters is easy with the CAL program. Run CAL, first turning the potentiometer all the way to its lowest setting and take a measurement. Then do the same at the maximum number of turns and save the CAL results to an .ini startup file.



Figure 10.9: 10 turn potentiometer, reverse side connections

How well do ratiometric methods work to reduce the problems of temperature drift? With the above setup, the relative voltage from the potentiometer as read by the SER1CH-UA does not vary by more than 30 microvolts (???) in our experiments. For quantitative results see the plots in the *Ratiometric measurements* chapter.

Potentiometers are useful for measuring angular position. However, they are only one class of sensors suitable for ratiometric use. For linear measurements over small distances, there are linear slider potentiometers. Figure 10.10 shows one example. Using a linear potentiometer is the same as a 10 turn rotational device. Connect the three terminals of the linear potentiometer up to the three terminals of the SER1CH-UA and you are ready. Calibrate the linear potentiometer into physical length units with the CAL program. For linear measurements over greater distances, there are also string potentiometers.

For passive ratiometric applications, there are also sensors such as strain gauges, tilt meters, and pressure transducers that all connect naturally to a three wire interface.

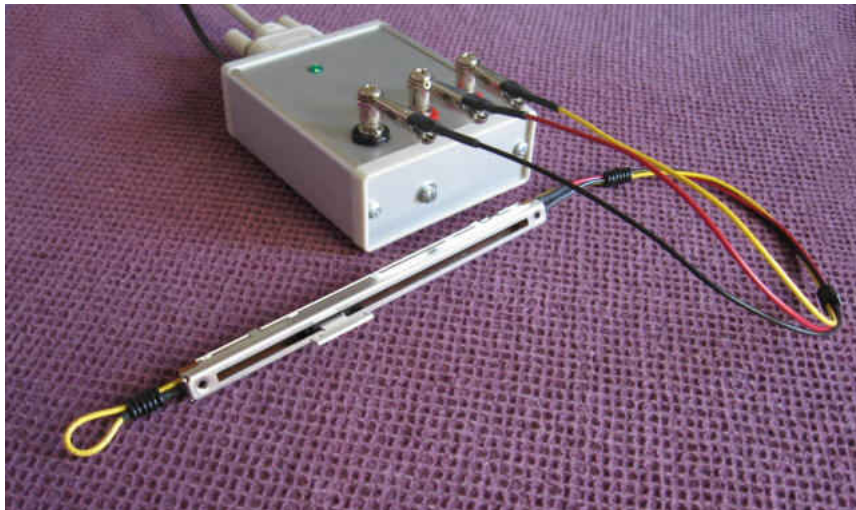


Figure 10.10: Linear slider potentiometer

10.4 Scaling and biasing +/-10 volt signals into (0,5)

The SER1CH-UA has a full scale input range of (0,5) volts as measured with respect to its red VIN and black GND banana jacks. However, for many applications you may require a different full scale range. In particular, (-10,+10) is popular for many active sensors.

Scaling and biasing a (-10,+10) signal into the (0,5) range is fairly easy with a few resistors. This demo shows how to preform the mapping with four 10K ohm resistors. More details about scaling and biasing can be found in:

- the *Scaling active sensor voltages* chapter of this User Manual for other voltage ranges
- for additional theory, the SR application note, *Scaling and Biasing Analog Signals*

The demo circuit for (-10,+10) mapping is as in Figure 10.11:

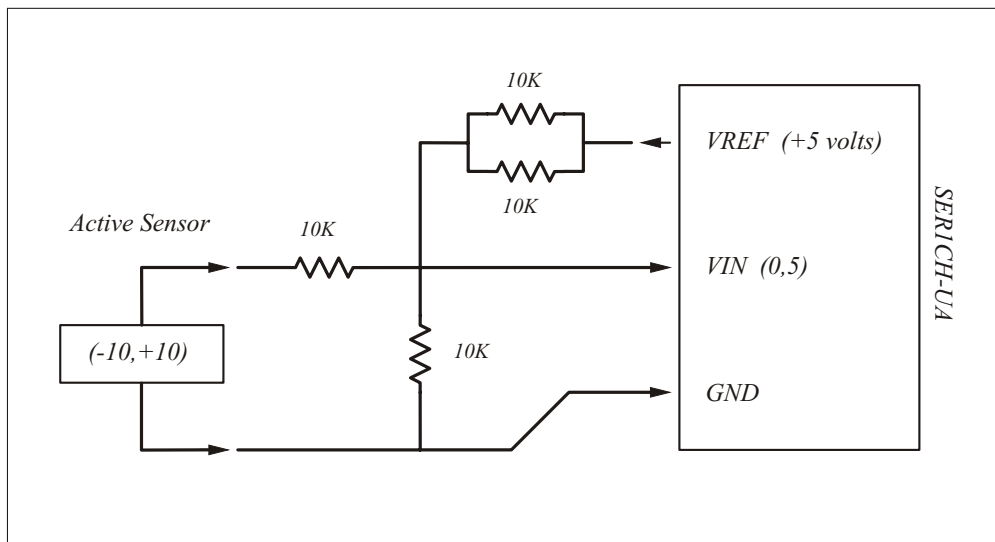


Figure 10.11: Scaling and biasing (-10,+10) with four 10K resistors

This circuit is the same as the one marked with a ★ star in the Figure 9.3 table in the *Scaling active sensor voltages* chapter. For this demo the 5K biasing resistor is built from two 10K resistors in parallel, since that may be all that is on hand.

There are several construction techniques for building circuits of this type. Here, we will use vector perf board, resistors with wire leads, wire, and solder. The popular white prototype punch boards are not recommended. The resistance at the punch board spring clip contacts varies considerably and is subject to corrosion. Simple solder joints result in a more reliable and stable circuit. Use good quality 1/4 watt metal film resistors for best performance.

Build the three resistor +/-10 to (0,5) divider on vector board to demonstrate one construction technique. Review soldering resistors together. Include JPEG photo. Suitable resistors are listed in the extra supplies section.

Mention resistor values can often be built using series or parallel combinations of a single stock value. So for example, with the +/-10 three resistor divider, use 4 10K resistors total, with 2 in parallel to build the 5K.

Demonstrate the input/output mapping with the scope and Gnu plots?

Make a point that a FLOATING input will not map to the 2.5 midpoint. The input must be grounded for the output to be at 2.5 ... an example that the item driving the input must be low impedance.

Also discuss where there are pads on the RevB board for smt parts. Maybe include a JPG for this too.

10.5 Measuring temperature

Reasonable temperature measurements can be made using inexpensive silicon diodes as the sensor. This example shows how to use a small signal 1N4148 diode to build a temperature sensor.

All silicon pn junctions have a change in their forward voltage of $-2 \text{ mV}/^{\circ}\text{C}$ with changes in temperature. This TC parameter is governed by the quantum mechanics of a pn junction and is remarkably constant across different diode lots. The total forward diode voltage depends on the amount of current flowing through the diode and will generally be around 600 to 700 mV. For any given diode, if the current through it is held constant, then its forward voltage will stay the same with only the TC variation remaining. When used in this way it becomes a reasonable temperature sensor.

The SER1CH-UA can easily measure changes of -2 mV in voltages of 600 to 700mV, and its DVM program can be calibrated to display the result as a temperature. The key is to maintain a constant steady current through the diode.

A rudimentary circuit can be as simple as:

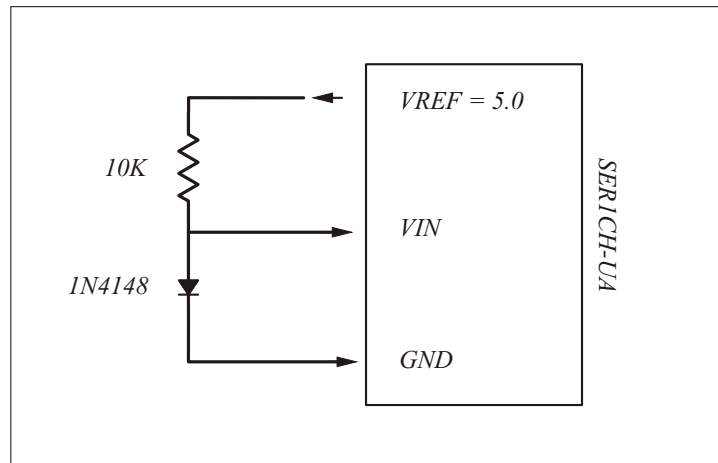


Figure 10.12: Temp diode with simple resistive current source

In Figure 10.12, the SER1CH-UA VREF is used as a 5 volt source for the diode. To a crude approximation, if the voltage across the 10K resistor is taken to be 5 volts, then an approximately constant current of 0.5 milliamps will be flowing in the diode. Of course, the current is only approximately constant because as the forward diode voltage changes with temperature the voltage across the 10K resistor changes also. Nevertheless, for many applications this is may be ok.

With the addition of an op amp, a better constant current source can be implemented as in Figure 10.13. For this circuit, the SER1CH-UA VREF is divided down to 1 volt for the noninverting op amp input. By negative feedback the inverting terminal is also forced to be at 1 volt, and this 1 volt puts a constant current down the 10K resistor connected to the inverting terminal. Since no current flows into the inverting terminal of the op amp, the same constant current flows through the

diode. The output of the amp will be at 1 volt plus the forward diode drop.

For this circuit, an LM358 or LM324 is a good generic op amp to use. These devices can even be powered in single supply fashion by VREF itself. Consuming only about 1 milliamp of power, their power requirements will not overload the VREF output. No split op amp supply is required.

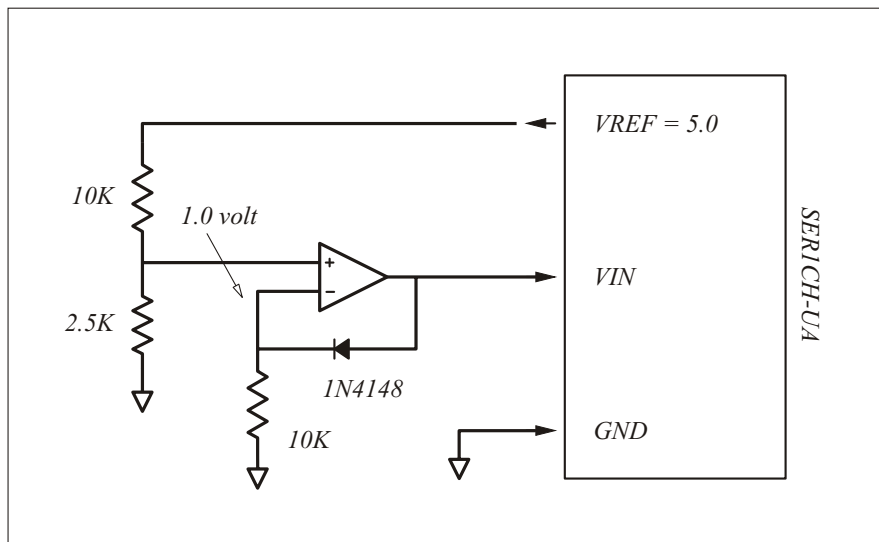


Figure 10.13: Temp diode with constant current op amp source

There are a wide variety of silicon diodes suitable for use as temperature sensors. The 1N4148 or 1N914 are probably the least expensive, costing as little as \$0.05 in qty 1. See the *Extra supplies* chapter in this User Manual for a few vendors. Solid state sensors optimized for temperature measurement are also available. Visit the web sites of suppliers such as National Semiconductor, TI, or your favorite semiconductor company for many listings.

Calibrate DVM into °C or °F for a nice display on the computer screen of the temperature measured in this way ...

Chapter 11

Absolute DC calibration

Initial absolute calibration of the SER1CH-UA is performed at the time of production. The calibration numbers are included on a card supplied with the system.

However, you may wish to perform periodic absolute calibration yourself. To do so requires either a known voltage reference, or a reliable precision voltmeter such as the HP34401A.

Reference standards such as the Symmetric Research VREF-399 are suitable, calibration tools.

Chapter 12

Single ended grounding practices

The SER1CH-UA has a single ended input. Excessive currents flowing on the black GND banana pin will affect *absolute* measurement accuracy.

UNDER CONSTRUCTION ...

Review typical wire and copper trace resistances and the errors they introduce.

Chapter 13

Batteries and minimizing power

With its small current consumption, the SER1CH-UA is a good match for many low power applications. However, to achieve the lowest power possible there are a few details to be aware of. Not knowing these details may result in the SER1CH-UA consuming more power than you expect.

This chapter reviews the typical current consumption of the SER1CH-UA, followed by steps to minimize the current drain on power supplies. It concludes with a typical example of powering the system from batteries.

13.1 Measuring current consumption

The physical current consumed by the SER1CH-UA can be measured in a few different ways. If the unit is out of its enclosure, one way is to measure the voltage drop across the 5.1 ohm antisurge resistor R900 in the power supply area of the board. Then use Ohm's law to compute the current.

If the unit is in its enclosure, you have two options. One is to include an external series resistor in the power supply line. Any resistor with a few ohms to a hundred ohms resistance will suffice. The voltage drop then implies the current. Alternatively, you can use an ammeter in series with the power supply and read the current directly. It will require an accurate ammeter with 1 milliamp or better accuracy.

Once you measure the current consumption of your particular SER1CH, of course you'll want to compare it against the quoted specs. Figure 13.1 computes the theoretical maximum base power requirement for the SER1CH-UA. Usually not all these individual current components are consumed at the same time and the typical current consumption will be approximately 600 microamps as quoted in the specification table.

SER1CH-UA MAXIMUM BASE POWER CONSUMPTION ...

PART	DESCRIPTION	CURRENT CONSUMPTION IN milliamps
R112	47K @ 5v	0.10638
R115	47K @ 5v	0.10638
R122	1M @ 5v	0.00500
U110	74HC14 quiescent	0.04000
U200	LTC2400 A/D converter, while running	0.20000
R211	100K @ 5v	0.05000
U210	TLC27L2 quiescent	0.03400
U910	TLC27L4 quiescent	0.06800
R921	LM4040 with 24K source resistor, 2.5v / 24K	0.10417
R922	100K @ 2.5v	0.02500
R923	100K @ 2.5v	0.02500

Total base current consumption (in milliamps) = 0.76393

Figure 13.1: Maximum base power consumption

13.2 Minimizing power

If you are consuming more current than the amount listed in Figure 13.1, there are four possible offenders:

- The green power LED is on, this will consume 10 milliamps
- A low value resistance is connected between VREF and GND
- Custom software is not handling the RS232 port correctly
- The power supply is greater than 18 volts

Disconnecting the green power LED can be done by removing jumper J910 on the board. It may be hard to believe a simple green LED can consume over 10 times more current than the A/D circuit itself, but it is true. Removing the LED jumper will save that current. Having a visible power on indicator is handy in the lab, but extracts its price when running from batteries.

After disabling the green power LED, it is also easy to forget resistive loads connected between the VREF banana jack and GND will consume additional power. For example, if a 1K ohm potentiometer is connected between VREF and GND, that alone will consume 5 milliamps. And, that 5 milliamps will have to be supplied by the SER1CH-UA power supply. A 10K potentiometer is often a good tradeoff between avoiding noise and minimizing power.

The third item, handling the RS232 serial port correctly, is a concern only if you are writing your own custom software.

The problem is with RS232 inputs on the PC end of the DB9 cable and the impedance they present, typically 5K ohms. If such an input is driven at +5 volts by the SER1CH-UA then this will consume 1 milliamp, which ultimately must be supplied by the SER1CH-UA power supply.

The SER1CH-UA has only one input to the RS232 port, the CTS signal. Don't idle this signal high or else you will burn unnecessary current on the order of 1 to 2 milliamps. An easy way to avoid trouble is to assert the LTC2400 chip select signal (RS232 RTS) only when needed. When this signal is deasserted, then the CTS input will automatically idle low and not burn unnecessary power. The finished application programs supplied with the SER1CH-UA do this automatically.

The last item, using a power supply greater than 18 volts, is a serious error. Circuitry on the SER1CH-UA would be damaged by power greater than 18 volts and is protected by a TVS device. When the external power supply exceeds 18 volts, the TVS turns on and clamps the voltage to safe levels. Normally, the TVS is there to protect against power supply spikes. However, if the power supply voltage is simply too high, the TVS will equally well turn on and present a heavy load to your external power supply. Check your power supply equipment beforehand and *don't exceed 18 volts*.

13.3 Batteries

Once you have minimized the power consumption of the system, you may be interested in powering the SER1CH-UA from batteries. 9 volt batteries are attractive temporary power supply for this system, running over 100 hours. For longer term power, lead acid 12 volt batteries are useful.

UNDER CONSTRUCTION ...

Typical 9v battery power drain curves
Lowest supply voltage to maintain regulation
Power switches in front of the SER1CH-UA

Chapter 14

Frequently Asked Questions

The following FAQ may help if you have general questions about the SER1CH-UA. Here is a list of the questions for easy review:

Do you include source code ?
What format are DVM.out acquisition files in ?
Can I use the SER1CH-UA with Xcel spreadsheets ?
Can I use the SER1CH-UA with MatLab ?
Can I use the SER1CH-UA with Visual Basic ?
Can I use the SER1CH-UA with LabView ?
Does the SER1CH-UA have any digital buffering ?
What analog input voltage ranges are possible ?
Can I overdrive the VIN analog input ?
Do static shocks hurt the banana jack inputs ?
Why does waving my hand change the DVM voltage ?
Can I connect a 10 ohm resistor to the VREF output ?
Why is the system consuming more than 10ma ?
Is the SER1CH-UA connected to system ground ?
Is there any antialiasing filter ?
Can I set up the SER1CH-UA for 50Hz rejection ?
How can I change the rejection between 60 and 50Hz ?
Is twisted pair good for analog inputs ?
What kind of RS232 cable is required ?
What length RS232 serial cable can I use ?
Is the system opto isolated ?
Will USB to RS232 dongles work ?
Does the SER1CH-UA record GPS time ?

If you find a question of interest, then search on down the chapter for the answer.

14.1 Software

Do you include source code ?

Yes. The entire source code for *all* of the supplied software is included with all systems at no extra cost.

The software package including source code is also available for free download from our web site www.symres.com. You should download for review before purchase or to get the latest revision if you are a current user.

What format are DVM.out acquisition files in ?

ASCII with the data listed in columns. You can read the files in any text editor. The format is simple and can be imported into many other programs.

Two ASCII formats are possible, verbose and bare. Verbose has a header with information about the DVM acquisition parameters at the top. Bare format has just the data in columns.

Can I use the SER1CH-UA with Xcel spreadsheets ?

Import the DVM.out ASCII data files in bare format.

Can I use the SER1CH-UA with MatLab ?

A native MatLab format is not provided. Import the DVM.out ASCII files in the bare format.

Can I use the SER1CH-UA with Visual Basic ?

The underlying support library is written in C and is available as a DLL. It can be called from Visual Basic as well as other programming languages as long as you follow C calling conventions.

Symmetric Research does not provide Visual Basic programming advice for SER1CH-UA users writing their own programs.

Can I use the SER1CH-UA with LabView ?

Set up LabView VI interfaces to the SER1CH-UA DLL library with C calling conventions and it will work.

Symmetric Research does not provide LabView programming advice for SER1CH-UA users writing their own programs.

14.2 Hardware

Does the SER1CH-UA have any digital buffering ?

No, the SER1CH-UA does not have buffering memory for accumulating converted results. After a

conversion, the digital result is saved on the A/D chip. The PC must read that result. If another conversion is performed, the previous digital result is lost.

Other Symmetric Research A/D products feature sizeable FIFO buffers so the PC does not have to read results immediately. At the acquisition rates of the SER1CH-UA it is assumed the PC can keep up with the required digital bandwidth.

What analog input voltage ranges are possible ?

The analog input voltage may range from 0 to +5 volts. This is the voltage as measured on the middle red VIN banana with respect to the black ground GND banana. Valid count values are returned only for voltages within this range.

If you have another voltage range, then you must add external circuitry such as a voltage divider or amplifier to scale the voltage into the 0 to +5 range. See the *Scaling active sensor voltages* chapter for examples.

Many sensors can use the voltage reference provided on the VREF red banana jack to scale their ranges naturally into the full range of the SER1CH-UA. See the *Ratiometric measurements* chapter for examples.

Can I overdrive the VIN analog input ?

Input currents are limited by the signal conditioning circuitry on the SER1CH-UA front end. However, overdriving the input is hard on the system. We recommend adding suitable front end circuitry to deal with the particular problem at your site rather than making the SER1CH-UA take the brunt of the overload.

Zener diodes, TVS diodes, and gas arrestor tubes are all popular overload protection techniques depending on the size and magnitude of the problem.

Do static shocks hurt the banana jack inputs ?

Yes. Although the input circuitry has resistors and capacitors to help guard against static shocks on the banana jack connectors, static discharges are brutal to any electronics.

This includes small sparks generated from walking over a carpet. Static discharge events will result in a degradation of the electronics even if total failure does not occur.

Before touching the banana jack inputs, take a moment and discharge yourself by touching a nearby metal object such as a metal desk or a computer case. Then touch the metal case of the SER1CH-UA and then finally the banana jack inputs for any connections that must be made. The moment spent consistently performing this simple routine will help prevent static damage.

Why does waving my hand change the DVM voltage ?

The input impedance of the SER1CH-UA is 10M ohms. Waving your hand over the input banana jacks when nothing is connected will induce a measureable charge and voltage change by capacitive coupling. DVM reflects this change.

Connect the analog input VIN to the on board VREF or GND and you will notice that the effect

dissappears. This behaviour is typical of all high precision, high input impedance voltmeters. Floating inputs tend to wander as various charged entities are brought near. Connect the input to specific voltages to make accurate measurements.

Can I connect a 10 ohm resistor to the VREF output ?

No. Connecting a 10 ohm resistor across the 5 volt VREF and GND would theoretically result in a current of $5/10 = 0.5$ amp flowing! Obviously, this is more current than a device like the SER1CH-UA is designed to supply.

No damage will occur by connecting a 10 ohm resistor or even shorting VREF directly to ground. The VREF output is current limited and will supply an absolute maximum current of about 40ma. On our spec sheet lists a max of 10ma to stay within compliance. When overloaded the output voltage will sag and fall out of regulation.

If you need more than a few milliamps of reference current then use the VREF to drive an offboard buffer.

Why is the system consuming more than 10ma ?

You probably have the green power LED enabled. To achieve minimum power consumption the LED jumper must be removed. LED's typically consume around 10ma which is considerably more than the base power consumption of the SER1CH-UA.

See the *Batteries and minimizing power* chapter for more information regarding the LED and other power issues.

Is the SER1CH-UA connected to system ground ?

Yes. The black banana jack on the SER1CH-UA top panel is connected to the RS232 DB9 ground pin on the back panel which in turn is connected to PC ground.

In addition, the outer ring of the 2.1mm power supply connector is connected to the ground of the SER1CH-UA. If you use a grounded power supply, then the SER1CH-UA will also be connected to that ground. The wall transformer supplied with the system is an unregulated *floating* power supply.

Is there any antialiasing filter ?

There is no op amp antialias filter in the front end signal conditioning. However, the internal digital filter of the SER1CH-UA LTC2400 A/D converter has excellent 60 or 50Hz rejection. For low frequency measurements, powerline noise is the most common antialias filter requirement. See the *60 and 50Hz power line rejection* chapter for more information.

For rejection of higher frequency contamination, single pole RC filters are usually recommended. There are pads for installing a variety of RC combinations on the SER1CH-UA front end signal conditioning. See the *Circuit Diagrams* chapter for more information.

Can I set up the SER1CH-UA for 50Hz rejection ?

Yes. The internal filter of the LTC2400 used on the SER1CH-UA can be set to either 60 or 50Hz rejection, but not both. Changing the setting requires changing soldered on surface mount resistors on the SER1CH-UA board. We don't anticipate customers wanting to change these resistors much once they are working in their specific location. You can specify 60Hz or 50Hz rejection at the time of your order. If not specified, then the system will be set for 60Hz rejection.

How can I change the rejection between 60 and 50Hz ?

The selection of 60 or 50Hz rejection is done with 0 ohm jumper resistors soldered on the SER1CH-UA. To change the setting you will have to open up the SER1CH-UA enclosure and solder on the appropriate resistors. The resistors are 0603 surface mount parts.

By default, SER1CH-UA systems are shipped with the jumper resistors set for 60Hz rejection. You may request 50Hz at the time of ordering and your board will be configured appropriately. If no selection is specified, then the system was configured for 60Hz.

If you choose to change the 60/50Hz setting yourself, please refer to the circuit diagram in this manual and the Linear Tech LTC2400 spec sheet for additional details.

Is twisted pair good for analog inputs ?

Yes. Even though the SER1CH-UA sigma delta A/D converter has excellent 60 and 50 Hz rejection, it is better to avoid power line noise in the first place. Twisted pair for the input signals is one of the best ways to do so.

Long red and black test leads typically used with voltmeters create large open loops which easily pick up power line noise. Even simple steps such as loosely twisting the red and black test lead pairs together can substantially reduce power line noise pickup. Jacketed twisted shielded pair round cable is even better.

Foil shielding does not do much good against 60/50Hz noise. Foil shielding does help protect against static discharge damage and RF noise. At 60/50Hz it is the twist that does the job.

What kind of RS232 cable is required ?

Use a standard straight through DB9 Male to Female cable. Sometimes this is called a DB9MF extension cable. Do *not* use null modem cables. They swap signal wires and will not work correctly.

If you are using multiwire jacketed cable to make a custom cable, you will need 5 conductors. One each for ground, power, CTS, RTS, and DTR. We also recommend shielded cable for this connection, with the shield connected to the outer housing on the Dshell connectors on each end. The shield will conduct away any static discharges to the enclosures on each end.

Split out the power conductor and route it to the 2.1mm power connector. The other signals can go to the DB9 connector. If you are making a permanent installation consider using the SER1CH-UA-DSUB model which has all connections on Dshell connectors.

What length RS232 serial cable can I use ?

We have tried 100ft RS232 serial port cables. Signal quality of the digital signals was good and the system ran without trouble. Longer cables may work, but have not been tested.

Is the system opto isolated ?

No. The SER1CH-UA is connected to the ground and signal wires of the RS232 DB9. Those signals are not opto isolated from the analog ground of the SER1CH-UA.

Users requiring a floating system should use RS232 opto iosolator dongles that support CTS, RTS and DTR, and floating power supplies such as an unregulated wall transformer or a battery.

Will USB to RS232 dongles work ?

Only dongles that support the RS232 signals RTS, CTS, DTR can be used. Be careful, many dongles support only RX and TX.

One USB to RS232 dongle we have used successfully is from Keyspan. It is clearly advertised as supporting *all* the RS232 signals including RTS, CTS, DTR. Various other dongles have failed, so don't make assumptions.

Does the SER1CH-UA record GPS time ?

Not directly. However, the DVM application program is capable of recording the PC time along with each sample. If you have a GPS receiver or NTP time setting your PC clock, then that time will be recorded in the DVM.out files as a result. See the *Software* chapter for more details on DVM PC time stamping.

Note that time stamps derived from the PC clock give at best millisecond accuracy. This is prefectly adequate for many applications, particularly if they are sampling at the 1Hz rate. For more precise time stamping SR has other A/D converter products with accuracy in the sub microsecond range.

Chapter 15

Specifications

SER1CH-UA Specs					
Parameter	Conditions	Min	Typ	Max	Units
Resolution			24		bits
VIN input voltage range		0		5.0	V
Noise Floor (1)			10		μV
Input Impedance			10M		Ω
VREF voltage			4.8		V
VREF current				10	mA
VREF capacitive load				10	μF
Power Supply Voltage		8	9	16	Vdc
Power Supply Current (2)	converting		600		μA
Power Supply Current (3)	sleeping		200		μA
RS232 cable length (4)			6	100	feet
Operating Temperature		0		70	$^{\circ}\text{C}$
Dimensions			2.13 x 3		inches

Figure 15.1: Specifications

- (1) Peak to peak noise floor.
- (2) Continuous current when performing conversions, with LED off and no VREF load.
- (3) Sleep mode occurs when the LTC2400 CS pin is held high.
- (4) Absolute maximum RS232 cable length is yet to be determined.

Chapter 16

Circuit diagrams

FREE WEB VERSION - PARTIAL CIRCUIT DIAGRAMS

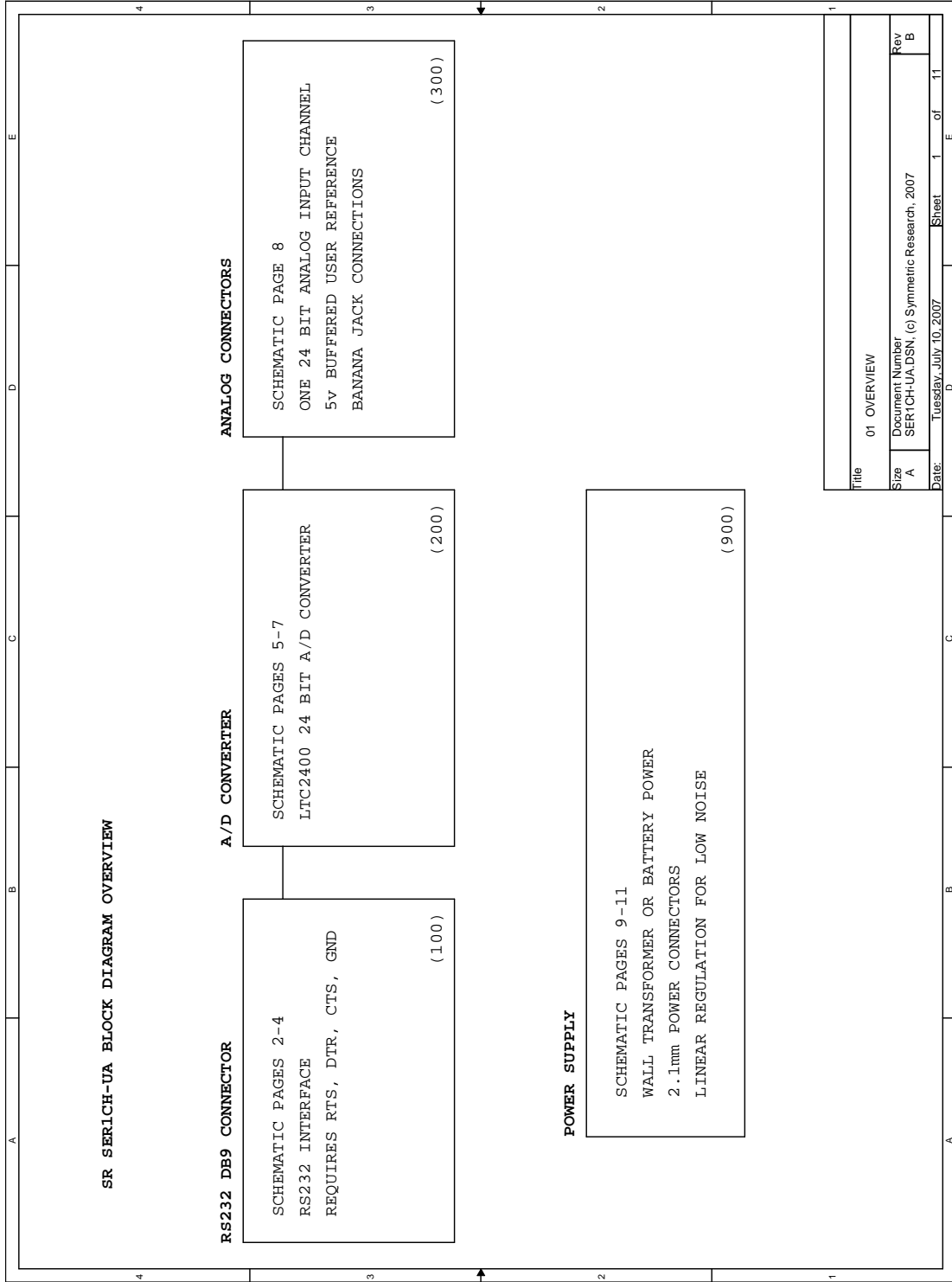
This free web copy of the User Manual includes only partial diagrams with the circuit Overview and the VIN analog input amplifier.

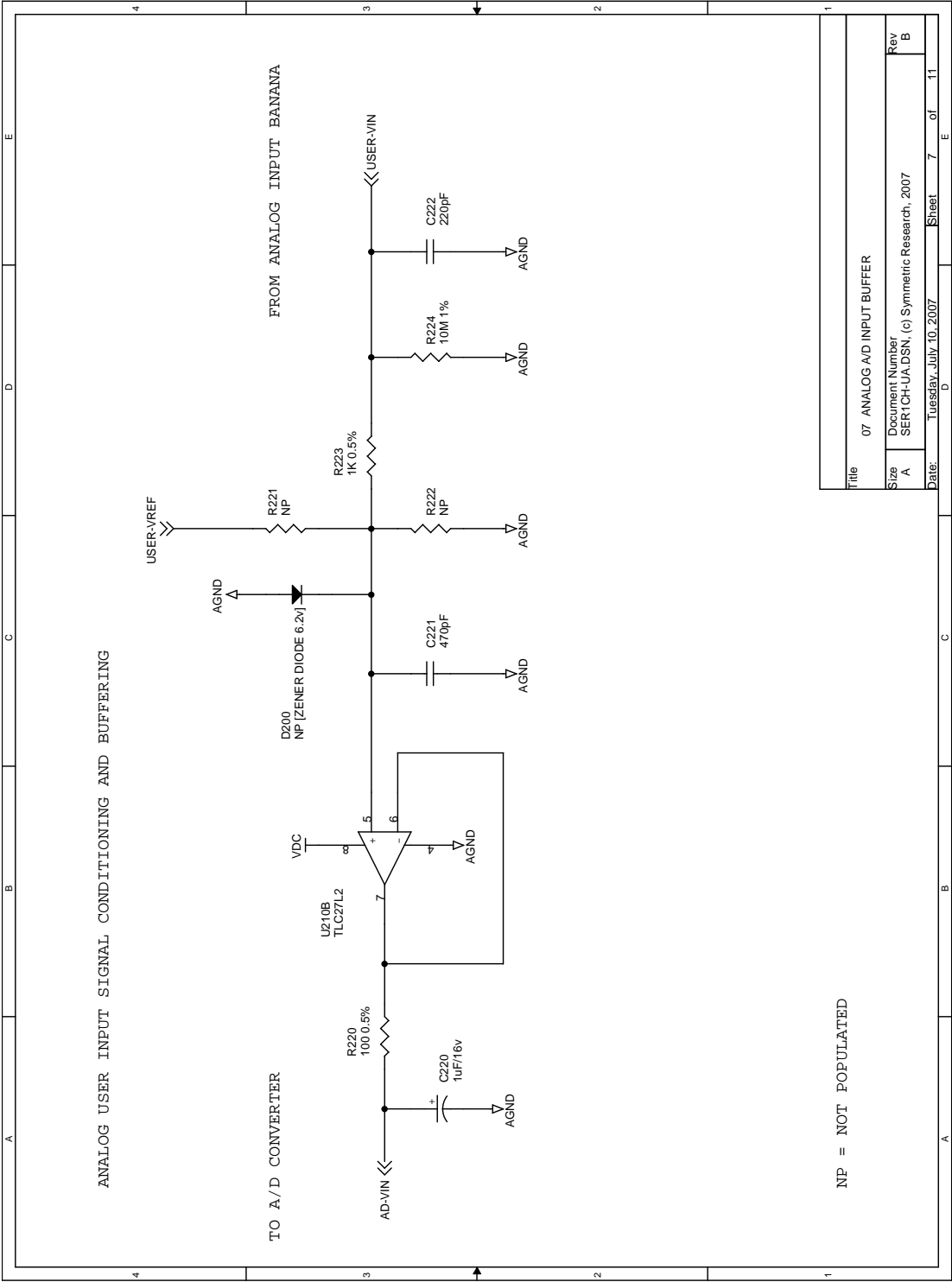
Complete circuit diagrams are provided with purchase of the product. Install the CDROM software included in the product shipment box for the full User Manual and complete diagrams.

Customers who need an update or have lost their original CDROM may obtain copies of the full User Manual by emailing SR at info@symres.com.

Use the Adobe Acrobat reader View/Rotate View/Clockwise command to present the diagrams horizontally on the screen.

- Overview 69
- A/D Input Buffer 70





Chapter 17

Extra supplies

A small collection of parts for getting started with the SER1CH-UA is included with all new orders. We will be glad to supply additional parts if you contact us.

The following list of mail order suppliers and part numbers may also be useful for obtaining small parts for use with the SER1CH-UA.

DigiKey
www.digikey.com
1(800)344-4539

Mouser Electronics
www.mouser.com
1(800)346-6873

JDR Microdevices
www.jdr.com
1(800)538-5000

Abbreviations used below:

MFG = Manufacturer
DK = DigiKey
MO = Mouser
JDR = JDR Microdevices

17.1 Small parts for cables etc

Some of the most commonly used small parts in SER1CH-UA applications are as follows. This is only a small listing of the parts and suppliers available. Many others are acceptable.

2.1mm power plug

MFG part number = CUI Inc PP3-002A
DK part number = CP3-1000A

MFG part number = Kobiconn
MO part number = 1710-2131

These are discrete wire power plugs suitable for soldering wires to. They are useful for connecting batteries and other custom power sources to the SER1CH-UA. Use these instead of chopping the plug off of your wall transformer.

Alligator clips

MFG part number = Mueller BU-30
DK part number = 314-1010-ND
MO part number = 548-30

The Mueller BU-30 clip has a good spring and strong teeth. The connection is wire crimp, which should also be soldered for low resistance. Making cables with these clips requires skill, but they result in inexpensive connections of good quality that are easy to use.

Johnson banana plugs

MFG part number = Johnson (Emerson) 108-753-001
DK part number = J149-ND
MO part number = 530-108-0753-1

These uninsulated banana plugs have a 6-32 screw in the end. When plugged into the red and black jacks on the SER1CH-UA, they provide screw terminal connections.

Solder terminal lugs for Johnson banana plugs

MFG part number = Keystone 7329
DK part number = 7329K-ND
MO part number = 534-7329

The Keystone solder lugs are useful for making screw terminal connections to the Johnson banana plugs. It is also easy to clip alligators onto these lugs when they have no wire soldered on. The lugs listed above are flat. Lugs with internal tooth lock washer holes are also available for even more secure connections.

Pomona test leads

MFG part number = Pomona 5519A
DK part number = 501-1004-ND
MO part number = 565-5519A

To use these test leads with the SER1CH-UA you must trim or cut away the banana plug insulating sheath. Do this with a utility knife or razor blade carefully cutting away the plastic sheath.

DB9 moulded serial port cables

MFG part number = generic
JDR part number = CBL-DB09-MF

Finished moulded 6' cable. Commonly referred to as a DB9-MF male female extension cable. All pins are connected straight through. Null modem cables will not work.

DB9 solder cup Dshell connectors

MFG part number = generic
JDR part number = DB09P (male)
JDR part number = DB09S (female)

For making custom RS232 cables, solder cup DB connectors are convenient. These are available everywhere including DK and MO. The JDR parts are generic and of good quality at a reasonable price. Since the RS232 cable is MF you need one for each end.

Wall transformers

MFG part number = CUI Inc DPD090020-P5P-SZ
DK part number = T967-P5P-ND (9v 200ma 2.1mm plug, 110vac US)

MFG part number = CUI Inc DPD090050E-P5P-SZ
DK part number = T973-P5P-ND (9v 500ma 2.1mm plug, 220vac Euro)

MFG part number = Xicon
MO part number = 412-109024 (9v 200ma 2.1mm plug, 110vac US)

Don't use switching regulated wall transformers with the SER1CH-UA. No damage will occur, but switching regulators will introduce considerable high frequency noise into the system. A common unregulated linear wall transformer is generally preferable.

Single twisted pair shielded cable

MFG part number = Belden 9501
MO part number = 566-9501-100

Twisted pair provides reasonable powerline noise immunity for analog inputs. This Belden cable also has a foil shield and is jacketed. There are a large number of cable types available. This is only one example.

Double twisted pair shielded cable

MFG part number = Belden 9502
MO part number = 566-9502-100

For connecting potentiometers three wires are needed. This is the double pair version of single twisted pair. Use one wire from each pair as ground, and the other wire of each pair as signal.

1% metal film resistors

MFG part number = Yageo many values
DK part number = (Value)XBK-ND

MFG part number = Xicon many values
MO part number = 271-VALUE-RC

Metal film resistors give good performance for scaling and biasing applications. They cost very little more than carbon film parts but have better TC and noise characteristics. The parts above are standard 1/4 watt types with wire leads. For most applications, precision greater than 1% metal film costs considerably more and may not deliver substantially improved performance. Hand matching 1% metal films is a reasonable way to obtain tightly matched pairs. Hand matched carbon films will not deliver the same performance.

Potentiometers, 10 turn, 10K ohm

MFG part number = Vishay (Sfernice) 53411103
MO part number = 594-53411103

MFG part number = Bourns 3540 series
DK part number = 3540S-1-103-ND

A 10 turn pot is a simple and reasonably accurate sensor to use when testing the SERICH-UA. Connect the three potentiometer connections to the three banana jacks with the wiper going to the red VIN jack and you can make accurate angular measurements. There are many manufacturers of 10 turn pots. A 10K ohm unit will not draw excessive current from the VREF output.

Potentiometers, linear slider, 10K ohm

MFG part number = Alps RSA0N11S9002
MO part number = 688-RSA0N11S9002

Linear slider pots do not have the accuracy of a 10 turn pot, but they are ideal for making linear position measurements. As with rotary pots you only have to connect the three terminals to the SERICH-UA and calibrate appropriately. The above pot has a linear travel length of 120 mm = 4.7 inches.

Sensors, general

DK part number = Many, see their sensors listings
MO part number = Many, see their sensors listings

Both DK and MO have extensive selections of sensors that are suitable for use with the SER1CH-UA. Pressure, vibration, and light sensors are just a few in their listings. They can often provide an easy solution to sensor selection. PDF copies of the DK and MO catalogs are available for free download from their web sites. For many simple measurements like position or temperature, often very simple parts like potentiometers and diodes can do a perfectly good job. See the appropriate part listings here and chapters in the User Manual.

Sensors, temp diodes

MFG part number = Fairchild 1N4148
DK part number = 1N4148FS-ND

MFG part number = Fairchild 1N4148
MO part number = 512-1N4148

Virtually any silicon pn junction may be used as a temperature diode. Even the most inexpensive small signal silicon diode does a reasonable job. The generic 1N4148 or 1N914 are probably the least expensive, although if you have something else in your spare parts box it will probably do as well. More expensive solid state temp sensors are available at many of the semiconductor companies like National Semiconductor and TI.

Op Amps

MFG part number = TI LM324N
DK part number = 296-1391-5-ND

MFG part number = ST Micro LM324N
MO part number = 511-LM324N

The selection of op amps is large, each device optimizing some particular parameter. Probably the most generic part available is the LM324. The LM324 has four op amps (a quad) in a 14 pin ic package. Its close cousin, the LM358 is the same op amp, but with with two op amps (a dual) in an 8 pin ic package. These traditional bipolar devices were originally developed by National Semiconductor, but are now second sourced by many. They offer good performance at very low cost for general applications, and may be used with either single or split power supplies from as low as +5v to as high as +32v.

Batteries, 12 volt lead acid

MFG part number = BB Battery BP1.2-12-T1 (12v 1.2Ah)
DK part number = 522-1007-ND

MFG part number = Power Sonic (12v 1.4Ah)
MO part number = 547-PS-1212

The batteries above are small lead acid batteries. A good point about lead acid is they are rechargeable. A 1.2Ah battery would have enough power to run the SER1CH-UA continuously for $1.2/0.0006 = 2000$ hours = 83 days.



Copper Tailings
Ruth Nevada, 2007

Copyright ©, Symmetric Research, 2007

No guarantee of suitability for any application is made with this document. All liabilities are the responsibility of the user.

The SR document nnnnn.rrr is copyrighted and can only be reproduced in whole or part with the written permission of Symmetric Research

Email: info@symres.com Web: www.symres.com