

Volume

1

A-WIT TECHNOLOGIES INC.

... a passion for execution ...

Using the C Stamp with the SelmaDAQ Utility

Version 1.0

A-WIT TECHNOLOGIES INC.

Using the C Stamp with the SelmaDAQ Utility

© A-WIT Technologies Inc.
Phone (800) 985-AWIT • Fax (800) 985-2948

Table of Contents

Notices	2
Getting Support	2
Getting Started	2
Sample Code	3
SelmaDAQ Spreadsheet	4
Plotting with SelmaDAQ	7
Graphing your Data	10
Interactive Bar Graph	12
Developing Your Own Programs and Projects	16
Question and Answer	16
Terms and Conditions	17

Introduction

Data acquisition software is ideal for receiving and sending data to a micro controller. Providing information on the spot without having the user stop to take specific points of interest, or sending information to the controller are just some of the luxuries the software offers. With the use of this software and the C Stamp, data can now be received more efficiently.

With the use of SelmaWare's SelmaDAQ, users will be able to view data sent from the C Stamp. This data will be placed within an Excel spreadsheet where users will be able to manipulate and view the data that they received. Some of the software features are listed below:

- Plot or graph data as it is received with Microsoft Excel
- Record up to 26 columns of data
- Mark data with real-time
- Read/Write any cell on a worksheet

Notices

CSTAMP™ and CSTAMP™ Related Hardware Products, Software Products and Documentation are developed and distributed by A-WIT Technologies, Inc. All rights reserved by A-WIT Technologies, Inc. A-WIT SOFTWARE OR FIRMWARE AND LITERATURE IS PROVIDED “AS IS,” WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL A-WIT BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY ARISING OUT OF OR IN CONNECTION WITH THE SOFTWARE OR FIRMWARE OR THE USE OF OTHER DEALINGS IN THE SOFTWARE OR FIRMWARE.

MPLAB C-18 and MPLAB C-18 Users Guide is reproduced and distributed by A-WIT Technologies, Inc. under license from Microchip Technology Inc. All rights reserved by Microchip Technology Inc. MICROCHIP SOFTWARE OR FIRMWARE AND LITERATURE IS PROVIDED “AS IS,” WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL MICROCHIP BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY ARISING OUT OF OR IN CONNECTION WITH THE SOFTWARE OR FIRMWARE OR THE USE OF OTHER DEALINGS IN THE SOFTWARE OR FIRMWARE.

Getting Support

If possible, please check the C Stamp website www.c-stamp.com under SUPPORT for any updates to documentation, changes, or notices that may have become available since your Installation CD was produced. If you continue to have any issues for which a solution is not found in the aforementioned website, please e-mail tech_support@a-wit.com for help.

Getting Started

What you will need for this example:

- SelmaWare’s SelmaDAQ
- Microsoft Excel version 2000 or later

- C Stamp 48-Pin Module
- BOL Kit
- Digital Thermometer Part # CS452000
- 9 Pieces of Copper Wire
- Fundamental understanding of Microsoft Excel

After you have downloaded and unzipped SelmaWare's SelmaDAQ, double click the file that was downloaded to start the install wizard. Follow the wizard along selecting the typical installation for the software.

After the installation is finished, you should now have a new folder on your desktop called 'SelmaDAQ Data Acquisition'. If you do not see this folder on your desktop, go to Program Files and locate SelmaDAQ. Inside you should see two files. The first step is to open up the SelmaDAQ Help file and read through it paying close attention to the 'Directives' section. This section will give you all the commands that are necessary to interact your C Stamp with the SelmaDAQ spreadsheet.

Next go to the CS310XXX (μ C 101) Reference Guide Manual and scroll down to Project 8, Digital Thermometer. This will show you how to set up your thermometer on the BOL.

Sample Code

The code uses the **TEMPSIN** and **TEMPS_F** to calculate the temperature in Fahrenheit and assign it to the variable T. Then, using the **WCftoa** command, the floating point T is converted into its ASCII string and assigned to buffer. The first **SEROUT** command is outputting the word 'DATA', this word is needed for the SelmaDAQ to recognize that the next string or number will be displayed on the spreadsheet. The next **SEROUT** is outputting the buffer variable, which corresponds to the temperature. The last **SEROUT** command is to prevent a constant input of numbers. Without the "\n\r" characters, the software will not properly read the data. If you read the Help file you will notice that it states that all lines sent must end in, 'CR'. With using "\n\r", we achieve the same kind of response.

Note: A baud rate of 4.8K, with 8 data bits, and no parity is used. Also a single LED is lit to signify that the program is on or off. A delay of two seconds is used separate data being sent.

```
#include "CS110000.h"

void main(void)
```

```

{
// TYPE YOUR CODE HERE AFTER THIS LINE
RAM BYTE msg[] = "\n\r";
RAM BYTE test[] = "DATA,";
int interval;
RAM BYTE buffer[5];
float T;
float baudr = 4.8;

interval = 2000;

STPIND(46, HIGH);

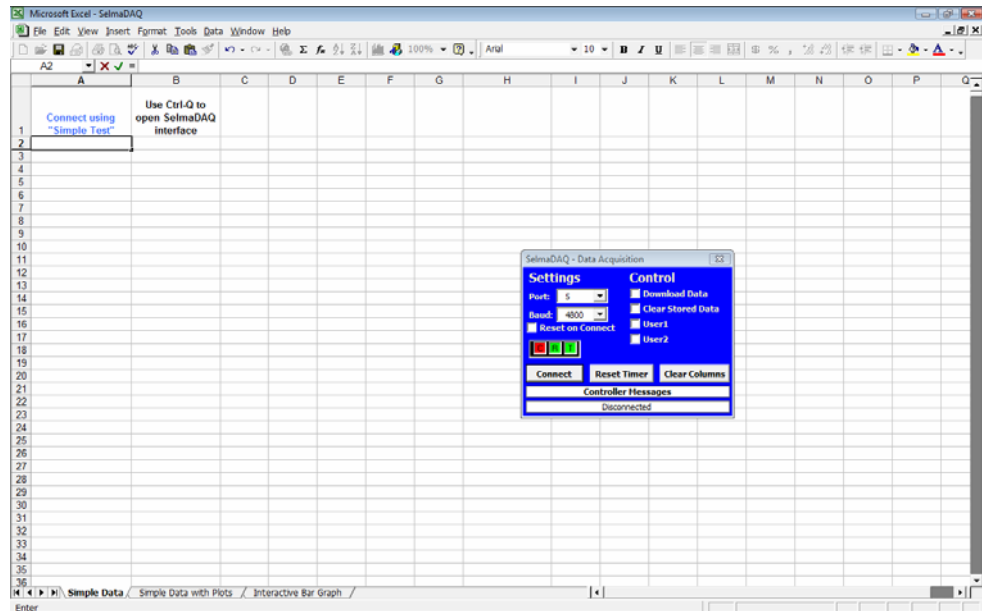
while(TRUE){
    T = TEMPSIN_CS452000(0, TEMPS_F);
    WCftoa(T, buffer);
    SEROUT(0, 0, baudr, 0, 8, 0, 0, test, 5);
    SEROUT(0, 0, baudr, 0, 8, 0, 0, buffer, 5);
    SEROUT(0, 0, baudr, 0, 8, 0, 0, msg, 2);

    PAUSE(interval);
}
}

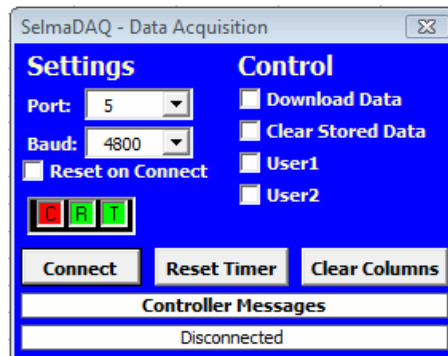
```

SelmaDAQ Spreadsheet

Start up the SelmaDAQ spreadsheet if you have not done so already. When the spreadsheet loads, you should see the following.

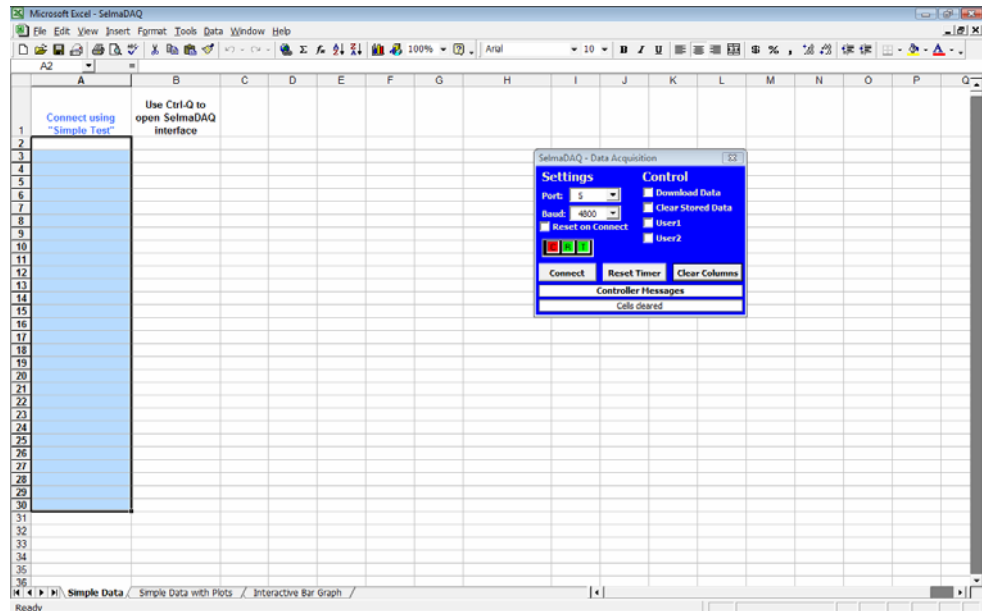


The next step is to connect the C Stamp to the spreadsheet. To do this, we must take a closer look at the SelmaDAQ interface shown below.

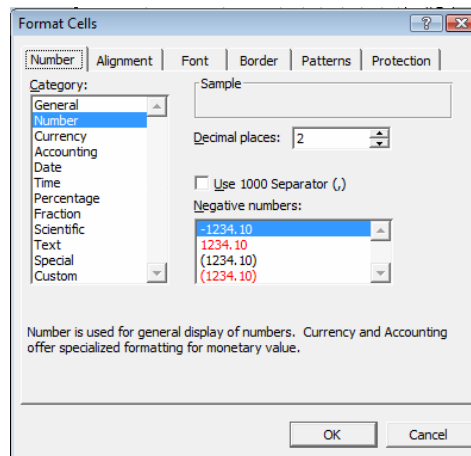


From here all of the settings, that will be required to receive or send data, can be set. Replicate your window to the one above with the exception of the COM Port. The COM Port should be set to the one you are using.

Next, for the purpose of this example only, we must change the properties of the first column. Select the first 20 to 30 rows as shown below, or as many as you will be using.

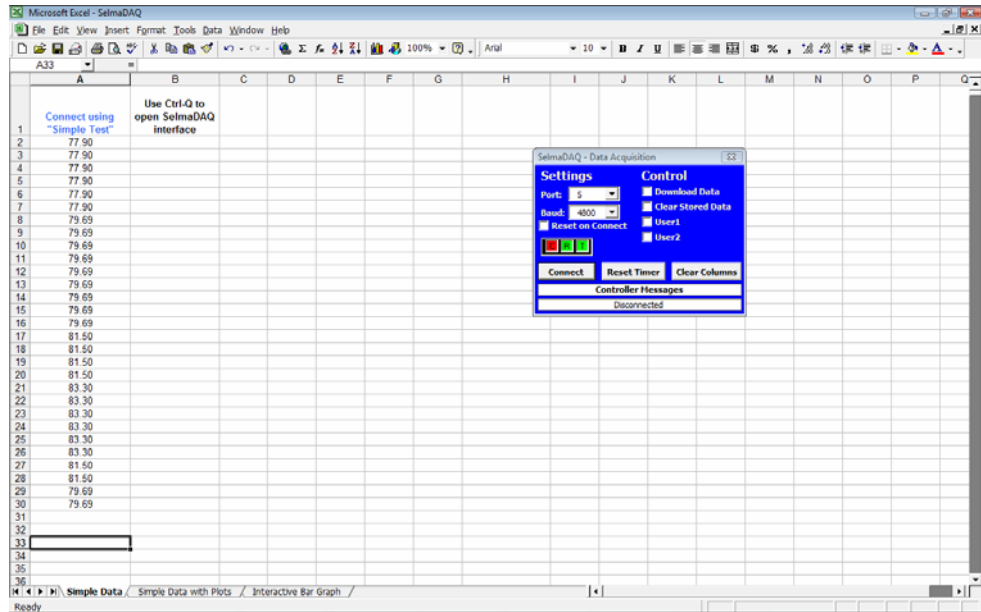


Right click on this column and select 'Format Cells'. Once the window pops up, select the 'Numbers' tab if it is not already selected.



In the Category select 'Number'. Choose the first option with two decimal places and press ok. This will now display the data in the first column as a number with two decimal places. If we did not make this change, the default settings of the first column are to display the data as a time. Without changing the settings, the data would not be displayed correctly.

When you have finished settings the configurations and column properties, connect to the C Stamp by pressing the 'Connect' button. If an error occurs, make sure that your configurations are correct. Once connected, if your program is running already, you should start to see temperature readings appear in the first column.



Plotting with SelmaDAQ

For this section, the code from the previous example will be augmented. To begin, either copy the code below into a new C Stamp project or add it into your existing code. The lines that are in italics and underlined below are the new lines of code that were added or changed.

The code will display the temperature as the last code did. The difference now, is we introduce the Directive 'TIMER'. This will keep track of the amount of time that has pasted in seconds since we were connected.

```
#include "CS110000.h"
```

```
void main(void)
```

```
{
```

```
// TYPE YOUR CODE HERE AFTER THIS LINE
```

```
RAM BYTE msg[] = "\n\r";
```

```
RAM BYTE test[] = "DATA,TIMER,";
```

```
RAM BYTE test_2[] = ",";
```

```

int interval;
RAM BYTE buffer[6];
float T;
float baudr = 4.8;

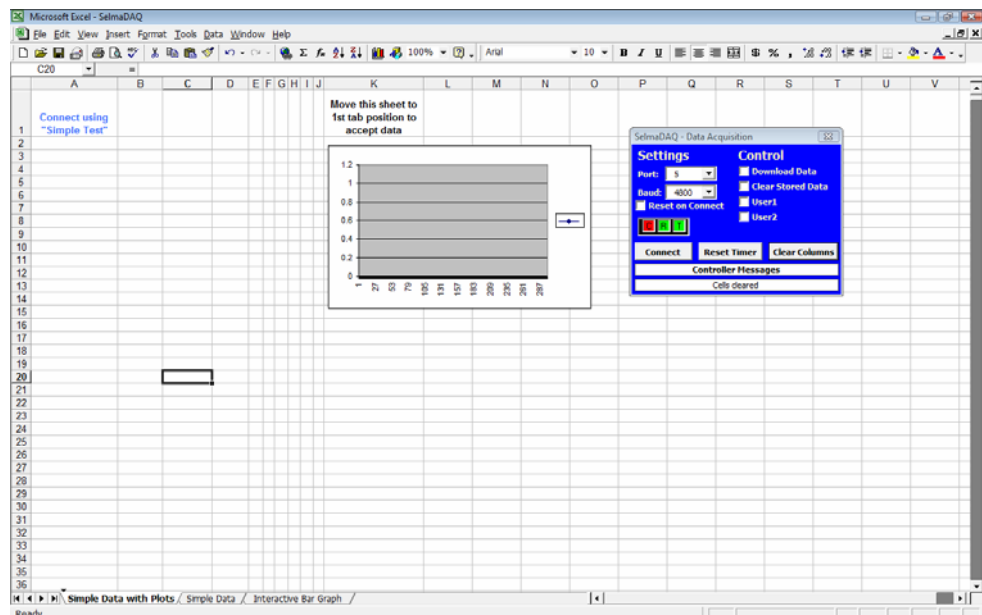
interval = 2000;
STPIND(46, HIGH);

while(TRUE){
    T = TEMPSIN_CS452000(0, TEMPS_F);
    WCftoa(T, buffer);

    SEROUT(0, 0, baudr, 0, 8, 0, 0, test, 11);
    SEROUT(0, 0, baudr, 0, 8, 0, 0, test_2, 1);
    SEROUT(0, 0, baudr, 0, 8, 0, 0, buffer, 5);
    SEROUT(0, 0, baudr, 0, 8, 0, 0, msg, 2);
    PAUSE(interval);
}
}

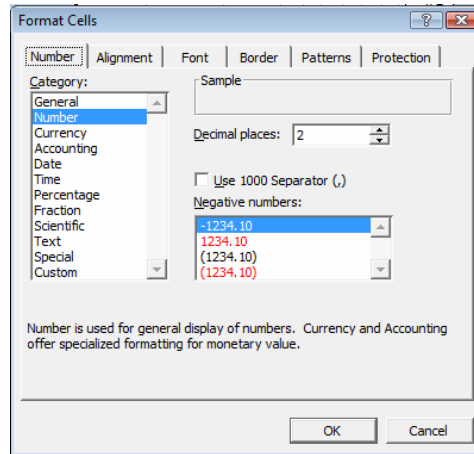
```

This code will be used within the ‘Sample Data with Plots’ sheet as seen below. We must place this sheet first in order for the data to be displayed on it. Go to where the spreadsheets are displayed and pick up the ‘Sample Data with Plots’ sheet and move it in front of the ‘Sample Data’ sheet.



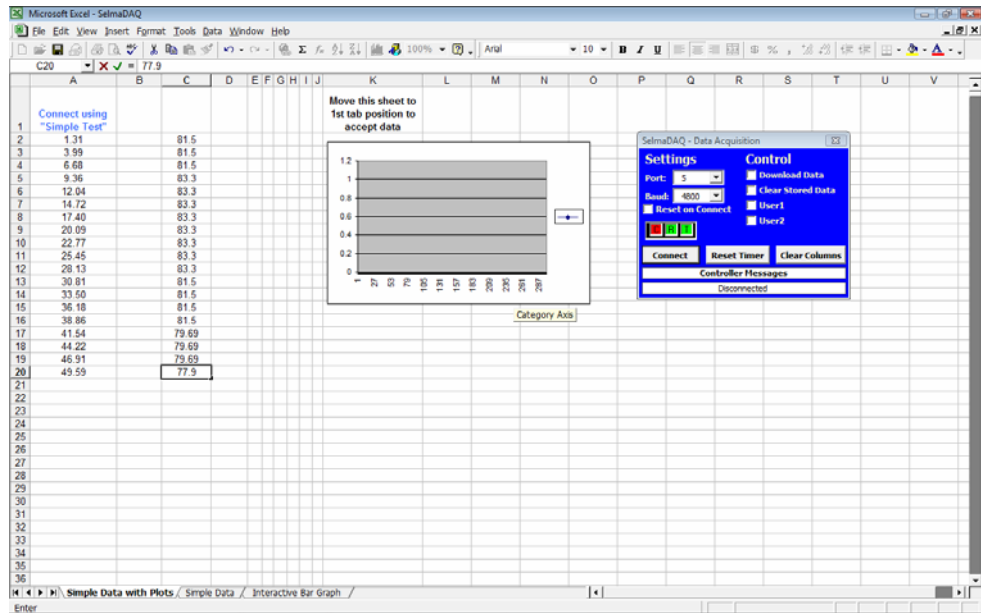
The settings for configuring the Data Acquisition should remain the same, but if they are not, view the previous section on instructions for their settings. The data that is received is a time in the first column and the temperature in the third column. In order to view the time correctly we must edit this column. Select the first 20 or 30 rows depending on how much data you wish to receive.

Right click on this column and select 'Format Cells'. Once the window pops up, select the 'Numbers' tab if it is not already selected



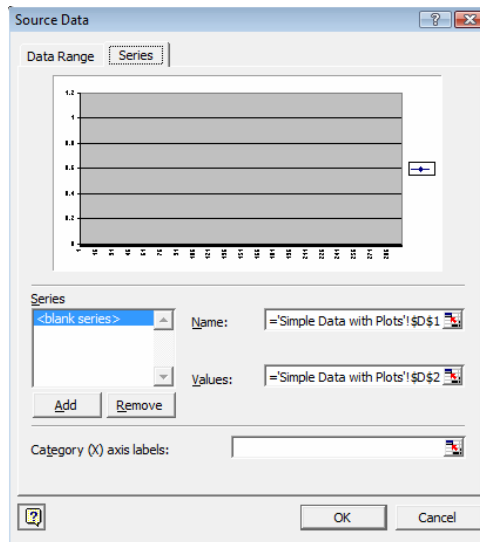
In the Category select 'Number'. Choose the default options with two decimal places and press ok. We must change the format of these cells for the purpose of this example. We are using the TIMER function; our data will be displayed in how many seconds the program has been running. If you want to display current time, change the call function TIMER to TIME.


When you have finished settings the configurations and column properties, connect to the C Stamp by pressing the 'Connect' button. If an error occurs, make sure that your configurations are correct. Once connected, if your program is running already, you should start to see time readings in the first column and temperature readings in the third column as shown below.

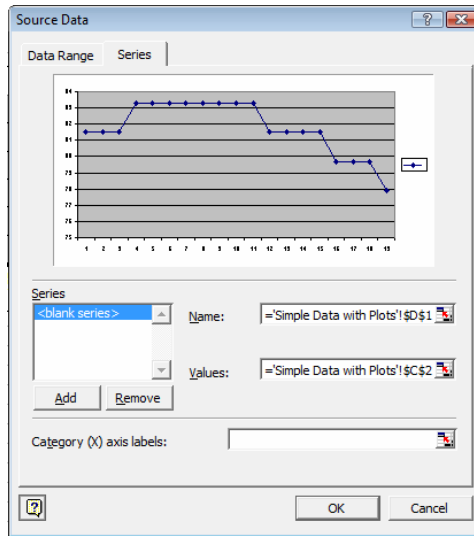



Graphing your Data

In order to graph the times and temperatures we must format the graph to the right. First, select the graph and right click it. Select the 'Source Data' option and select the 'Series' tab. A window should pop up as shown below.

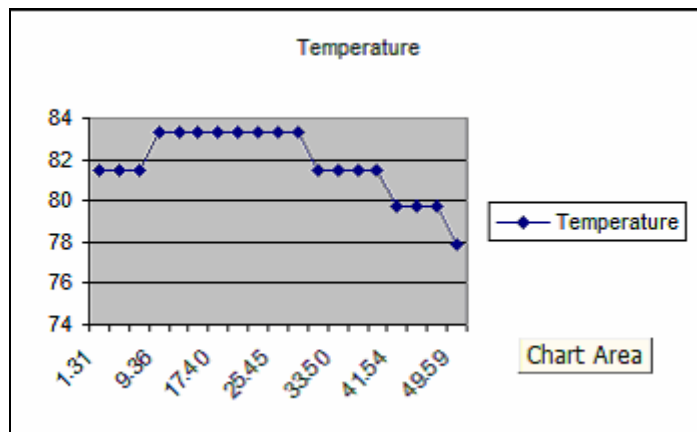


We want to first set up the Y-Axis to display out temperature values. In the ‘Values’ area click on the figure all the way to the right,  by clicking this will bring you back to the spreadsheet. Select all of the rows that contain your temperature readings and then press enter. After doing this, the preview of the graph should change to the following.



Next we will want to set up the X-Axis to display the times the data was taken. In the ‘Category (X) axis labels’ area, click the figure on the right.  This will once again bring you back to the spreadsheet. Select all of the rows that contain the temperature readings and then press enter.

Finally, we will name our graph. In the ‘Name’ box, clear the information currently there, and type Temperature. Press ‘Ok’ to return to the spreadsheet and to view our edited graph. If all the steps were done correctly, your graph should now be filled with your temperature readings and times.



Interactive Bar Graph

For this section, the code from the previous example will be augmented. To begin, either copy the code below into a new C Stamp project or add it into your existing code. The lines that are in italics and underlined below are the new lines of code that were added or changed.

This code introduces the use of more Directives. Here we will be introduced to the 'CELL,SET' Directive, and the 'ROW,SET' Directive. The 'CELL,SET' Directive will take data that it is assigned and assign the data to a specified cell. The 'ROW,SET' Directive will tell the spreadsheet to start displaying the data at that specified row.

The code below will place predetermined values into certain cells. From these cells, the graph will show the average temperature and the max temperature that was received. In addition, the total time that the program was running will be calculated.

```
#include "CS110000.h"

void main(void)
{
// TYPE YOUR CODE HERE AFTER THIS LINE
RAM BYTE msg[] = "\n\r";
RAM BYTE test[] = "DATA,TIMER, ";
RAM BYTE test_2[] = ", ";
RAM BYTE test_3[] = "ROW,SET,10";
RAM BYTE test_4[] = "CELL,SET,A6,Max Temp";
RAM BYTE test_5[] = "CELL,SET,A7,Average Temp";
RAM BYTE test_6[] = "CELL,SET,B6,=MAX(C10:C40)";
RAM BYTE test_7[] = "CELL,SET,B7,=AVERAGE(C10:C40)";
RAM BYTE test_8[] = "CELL,SET,A4,Total Time";
RAM BYTE test_9[] = "CELL,SET,B4,=SUM(A10:A40)";
int interval;
RAM BYTE buffer[6];
float T;
float baudr = 4.8;

interval = 2000;
STPIND(46, HIGH);
SEROUT(0, 0, baudr, 0, 8, 0, 0, test_4, 20);
SEROUT(0, 0, baudr, 0, 8, 0, 0, msg, 2);

SEROUT(0, 0, baudr, 0, 8, 0, 0, test_5, 24);
SEROUT(0, 0, baudr, 0, 8, 0, 0, msg, 2);

```

```

SEROUT(0, 0, baudr, 0, 8, 0, 0, test_6, 25);
SEROUT(0, 0, baudr, 0, 8, 0, 0, msg, 2);

SEROUT(0, 0, baudr, 0, 8, 0, 0, test_7, 29);
SEROUT(0, 0, baudr, 0, 8, 0, 0, msg, 2);
SEROUT(0, 0, baudr, 0, 8, 0, 0, test_8, 22);
SEROUT(0, 0, baudr, 0, 8, 0, 0, msg, 2);

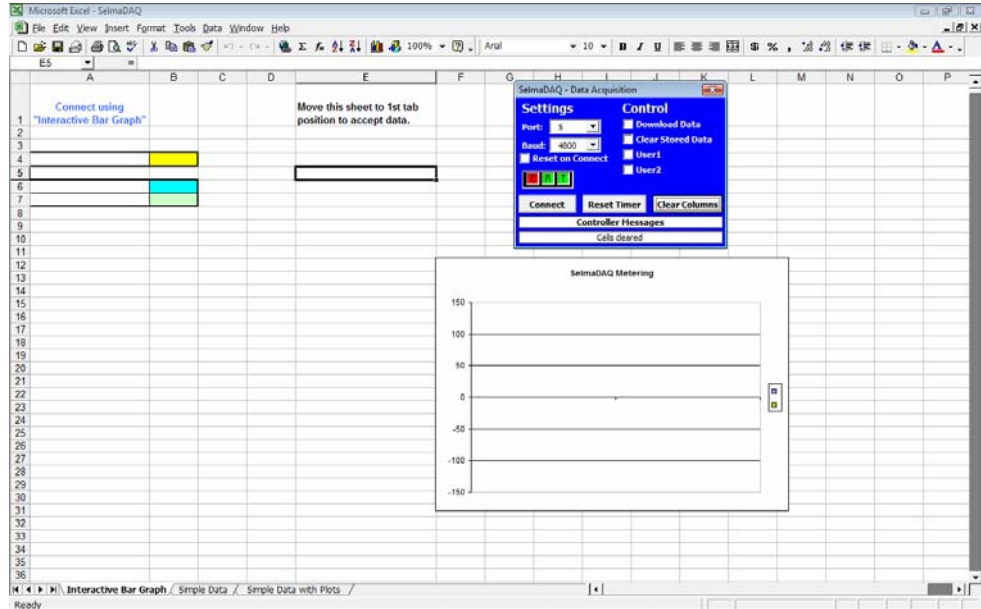
SEROUT(0, 0, baudr, 0, 8, 0, 0, test_9, 25);
SEROUT(0, 0, baudr, 0, 8, 0, 0, msg, 2);

SEROUT(0, 0, baudr, 0, 8, 0, 0, test_3, 10);
SEROUT(0, 0, baudr, 0, 8, 0, 0, msg, 2);

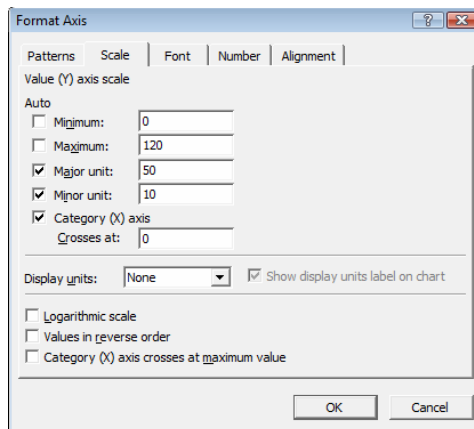
while(1){
    T = TEMPSIN_CS452000(0, TEMPS_F);
    Wcftoa(T, buffer);
    SEROUT(0, 0, baudr, 0, 8, 0, 0, test, 11);
    SEROUT(0, 0, baudr, 0, 8, 0, 0, test_2, 1);
    SEROUT(0, 0, baudr, 0, 8, 0, 0, buffer, 5);
    SEROUT(0, 0, baudr, 0, 8, 0, 0, msg, 2);
    PAUSE(interval);
}
}

```

This code will be used within the 'Interactive Bar Graph' sheet as seen below. We must place this sheet first in order for the data to be displayed on it. Go to where the spreadsheets are displayed and pick up the 'Interactive Bar Graph' sheet and move it in front of the 'Sample Data' sheet.

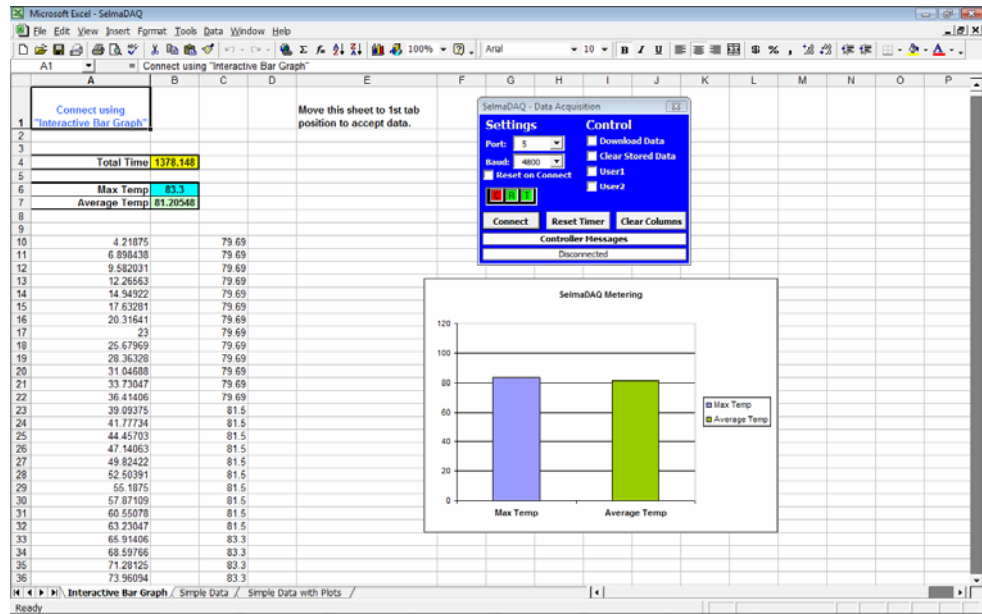


For this example configuration, the columns are not necessary. We will, however, configure the bar graph to suit our needs. Right click on Y-Axis. You will see an option that says 'Format Axis' select this option. When the window pops up, select the 'Scale' tab. From here, edit the 'Minimum' and 'Maximum' values to those as shown below.



When finished press ok, and now your graph should be configured properly.

When you have finished settings the configurations and column properties, connect to the C Stamp by pressing the ‘Connect’ button. If an error occurs, make sure that your configurations are correct. Once connected, if your program is running already, you should start to see time readings in the first column and temperature readings in the third column. In addition, the average temperature, maximum temperature, and total time will be calculated. The average temperature and maximum temperature will also be displayed in the graph.



NOTE: The code is only formatted to add and average rows 10-40. In order to change this, go to the code. Look for the following lines:

```
RAM BYTE test_6[ ]="CELL,SET,B6,=MAX(C10:C40)";
RAM BYTE test_7[ ]="CELL,SET,B7,=AVERAGE(C10:C40)";
RAM BYTE test_9[ ]="CELL,SET,B4,=SUM(A10:A40)";
```

To increase the rows that are recorded, simply edit C40 and A40 to how many data entries you wish to take.

Developing Your Own Programs and Projects

Now that you have successfully developed and run your programs, it is easy to move on to more complex and elaborate projects and circuits of your own. Remember though to read the Help file that comes with SelmaDAQ to understand what the program is expecting, in order for data to be received.

Question and Answer

The following section will try to answer the most common problems that can occur.

Q: Why am I receiving no data on the Excel sheet when I run my program?

A: NOTE: Remember in creating your own program, the SelmaDAQ must first see the string DATA for it to understand that you are sending data that is to be stored. Also you must end it with "\n\r" for it to go to the next line. Each new piece of information must be separated in order for the spreadsheet to read it properly. If you do not have this spacing nothing will appear on the sheet. Reference back to the sample code, and the Help file for information on the key words the spreadsheet is expecting. Also each piece of information must be separated by comma. A good way to know if you are sending the correct information is to view it in the HyperTerminal. If you are receiving:

DATA, (info), (info)

DATA, (info), (info)

then you are sending the correct information. Anything else will most likely not be read by the spreadsheet. As always, if you cannot resolve your problem feel free to contact our support team. For contact information, view the 'Getting Support' section of this manual.

Terms and Conditions

Quality Assurance

A-WIT has stringent quality control procedures in place to insure the best quality products.

90-Day Limited Warranty

A-WIT Technologies, Inc warrants its products against defects in materials and workmanship for a period of 90 days. If you discover a defect, A-WIT Technologies, Inc. will, at its option, repair, replace, or refund the purchase price. After 90 days, products can still be sent in for repair or replacement, but there will be a \$10.00USD minimum inspection/labor/repair fee (not including return shipping and handling charges).

14-Day Money-Back Guarantee

If, within 14 days of having received your product, you find that it does not suit your needs, you may return it for a refund. A-WIT will refund the purchase price of the product in the form of a check, excluding shipping/handling costs, once the product is received. This refund does not apply if the product has been altered or damaged. If you decide to return the products after the 14-day evaluation period, a 20% restocking fee will be charged against a credit.

Disclaimer

Warranty does not apply if the product has been altered, modified, or damaged. A-WIT makes no other warranty of any kind, expressed or implied, including any warranty of merchantability, fitness of the product for any particular purpose even if that purpose is known to A-WIT, or any warranty relating to patents, trademarks, copyrights or other intellectual property. A-WIT shall not be liable for any injury, loss, damage, or loss of profits resulting from the handling or use of the product shipped.

How to Return a Product

When returning, you must first e-mail sales@a-wit.com for a Return Merchandise Authorization number. No packages will be accepted without the RMA number clearly marked on the outside of the package. After inspecting and testing, we will return your product, or its replacement using the same shipping method used to ship the product to A-WIT within 30 days. In your package, please include a daytime telephone number and a brief explanation of the problem.

Please contact our Sales Department at sales@a-wit.com if you have any questions regarding our warranty policy or if you are requesting an RMA number.

