Customizing Automation in the Calibration Laboratory

Speaker/Author

Tom Lipko
BAE Systems Controls
600 Main Street
Johnson City, NY 13790

Abstract
I think the majority of calibration personnel would agree that automation in the laboratory is a tremendous asset. Some of the benefits of automation would be removal of operator bias, the ability of the technician to multitask and the ease of data collection just to name a few. These, along with other non-tangible assets, lead to cost savings and better efficiency. Of course these cost savings and efficiency come at a price, so the pay back time of the purchased software and software maintenance contracts plays a role in determining these benefits. There are a number of manufactures and software companies that produce calibration automation software but what do you choose? This was my dilemma. This paper describes my solution to this dilemma.

Introduction
I have looked at a number of automation packages. Some are very custom and specific and only deal with that company's calibration standards, so this limits you on what actually can be calibrated with that package. Another package offers you programmability but this is limited to current drivers that are supported. I have always dreamed of having a calibration software package that would allow me to create a text file of an instrument's test aspects and accuracy, have the application read the text file and calibrate my instrument. Since I would probably be waiting for this for along time, I decide to take on this task myself.

Requirements
So what do I want this automated calibration package to do? I came up with the following short list.

- **Ease of Use- Including both execution and procedure development**
    - **Graphical User Interface**
    - **Simple user procedure development**

- **Control Communications calibration standards during a single calibration**
    - **GPIB-IEEE-488**
    - **RS232**

- **Calculate Test Accuracy/Uncertainty Ratio**

- **Full Automation and Semi Automation Capability**
    - **Unit Under Test Control**
        - **IEEE-488 Communication**

        - **RS-232 Communication**
    - **Manual Instruction Control**

- **Standardize Data Collection**
    - **Future SPC Possibilities**

Development
A development platform had to be chosen to create the executable application along with an easy to use test file generator. Since I had written some prior applications in MS Visual Basic® I decided that this platform would be used to generate the application or what some call the test executive. The decision on the test file generator took some time. I needed an environment that users would feel somewhat comfortable during procedure development. Operating with a standard text file would require the user to be diligent when entering data since the data would have to be in the same format every time a procedure was generated. This would introduce too much error checking when loading the data file. In order to prevent loading errors, this strict format would have to be controlled. I finally decided to go with MS Excel ®. Some of the reasoning used to select the spreadsheet as the test file generator include overall restricted format entry, cells can accept both text or number entry , the application provides extensive math capabilities, it also provides multiple workbooks in the same spreadsheet and can be password protected. Another plus is that MS Visual Basic ® and MS Excel ® are able to talk to one another relative easy.

Test Executive
The GUI for the test executive is very simple. It provides four main GUIs for the user. The first GUI is a listing of the available calibration procedures as shown in Figure 1. This list is dynamic to the point that it lists all of the files with the .xls extension in the calibration directory. The purpose of this is to simplify the addition of new procedures. After the spreadsheet procedure is created it is simply added to the directory and appears as a new button when the test executive is run again.

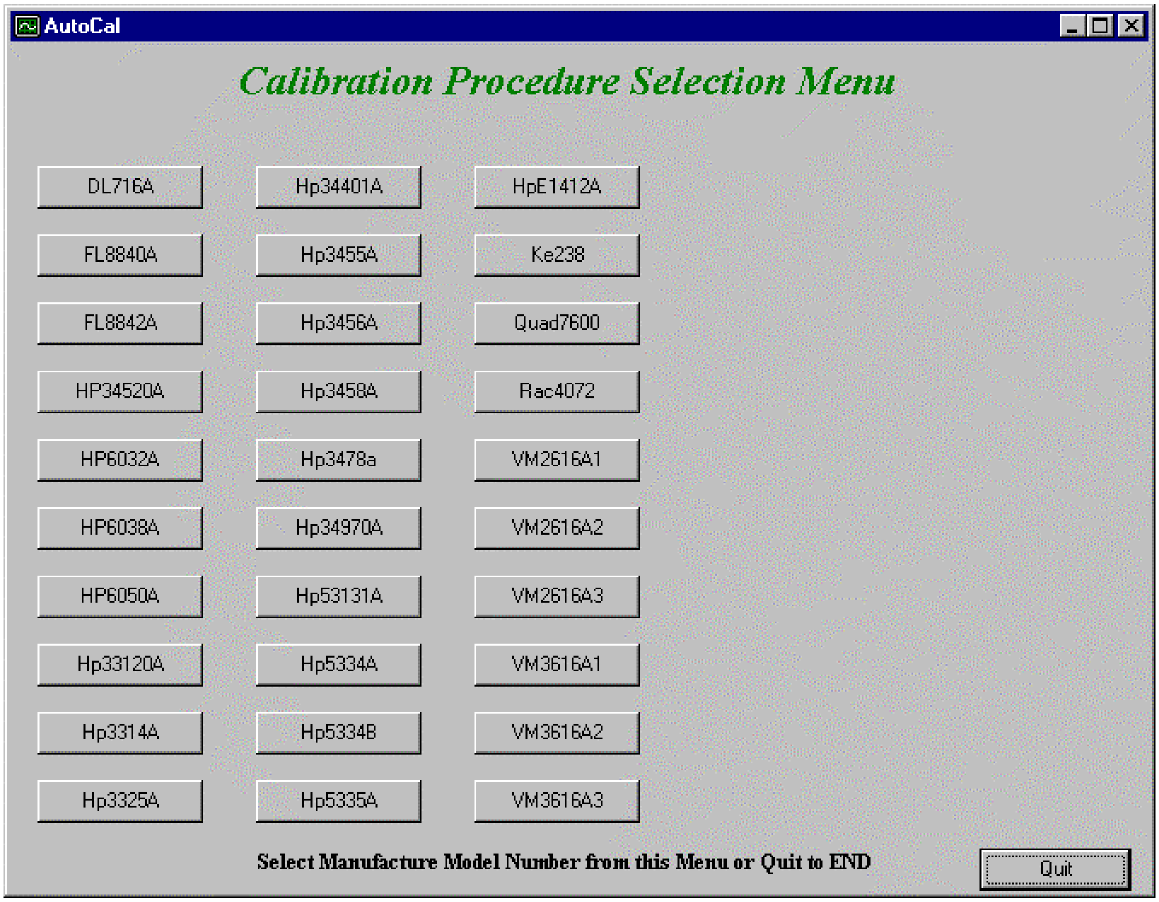


Figure 1-File Selection

Once the user selects which procedure to run, the database that contains all of the calibration information is loaded in to the test executive. Figure 2 shows the "Main Menu" of the test executive. There are some processes that are invisible to the user such as data file creation, IEEE-488 bus search for any listeners on the test bus and data file creation.

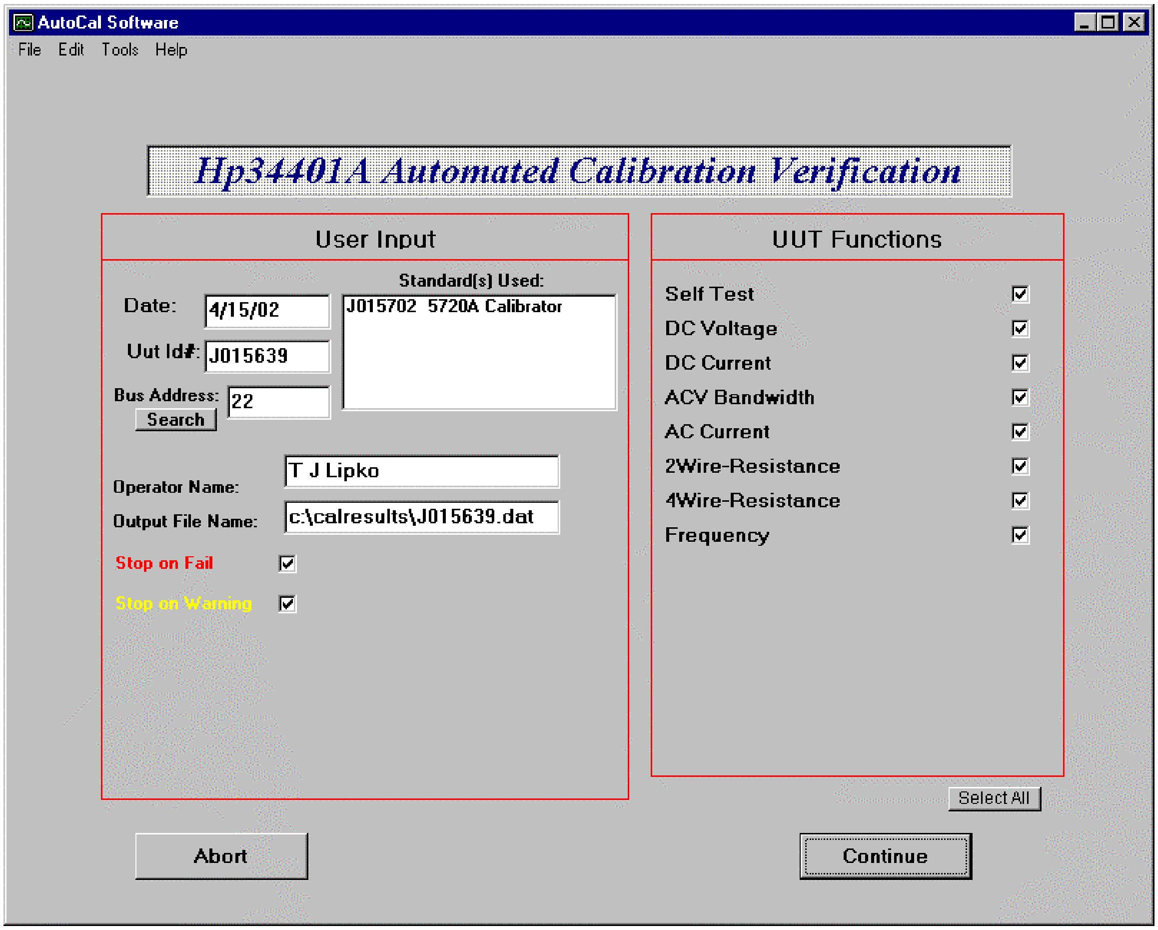


Figure 2-Main Menu

Any or all of the UUT functions can be selected. This option was created for the ease of verifying a single aspect and not having to execute the complete calibration. For example, if a UUT has a limited condition. The data file will reflect that the complete verification was not performed. Once all of the information is deemed correct, the user then selects to continue with the verification.

At this point the connection file or picture in viewed by the operator. There are two options here, the first is displaying a text file that instructs the operator how to perform the connection or second, a picture bitmap of the test connection is displayed.

Figure 3, represents the data screen. This interface gives the operator a real-time view of what is going on between the software, the standard(s), and the UUT. The aspect being verified is displayed along with the number of tests for that function. The operator has to options to stop the test if there are failures or the error is beyond the "guard-band" percentage. At this point the operator can see if there are problems with the test setup.

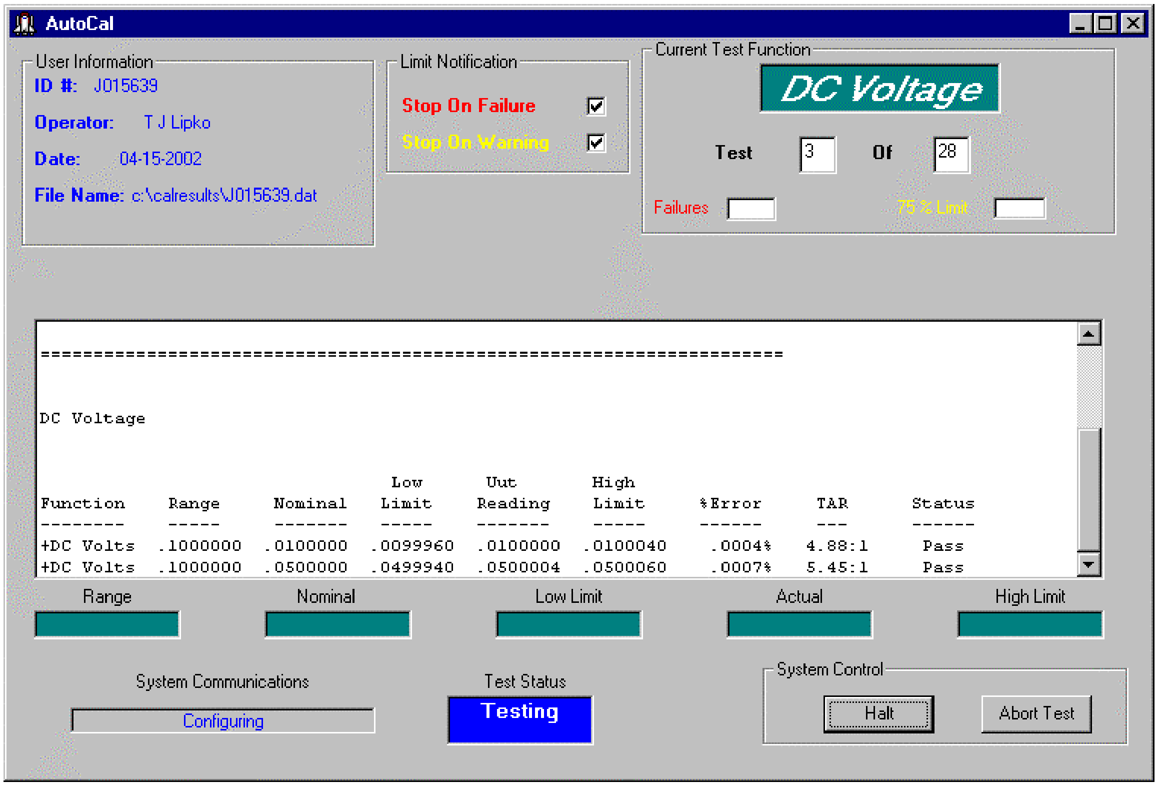


Figure 3- Data Screen

This screen displays the system communications between the software application and associated instruments. The test status indicator shows three levels of test status, Blue-test-in-process, yellow-guard band limit, and red – aspect failure. The system also allows the user to select how many "retries" are done before accepting the results.
The system control buttons allow the operation to halt the test and behind the "Abort Test" button are some system control utility options that allow communication control, database reload, and program termination. This screen is displayed until all of the selected aspects are tested. Once this has been completed, the summary screen is presented.

The summary screen, Figure 4, displays all of the aspects that were tested, the results of the tests and the option to print the summary or launch a text editor to view the data file. The data file is generated using the ID number asked for at the beginning of the test. If this is a repeated test of the same ID number then this new data is appended to the old data file. I would like to create a SPC tool in future versions to review this data since unlimited history can be stored.

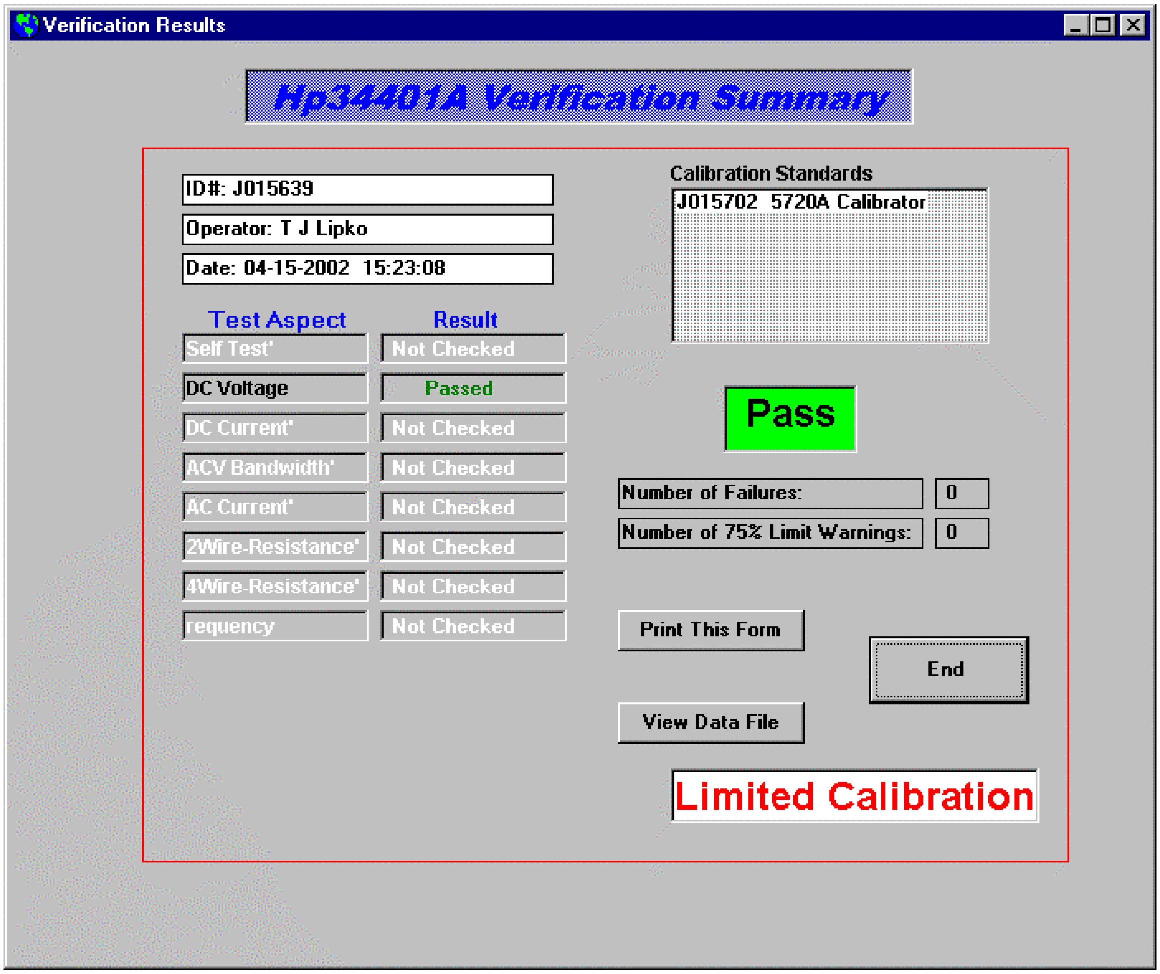
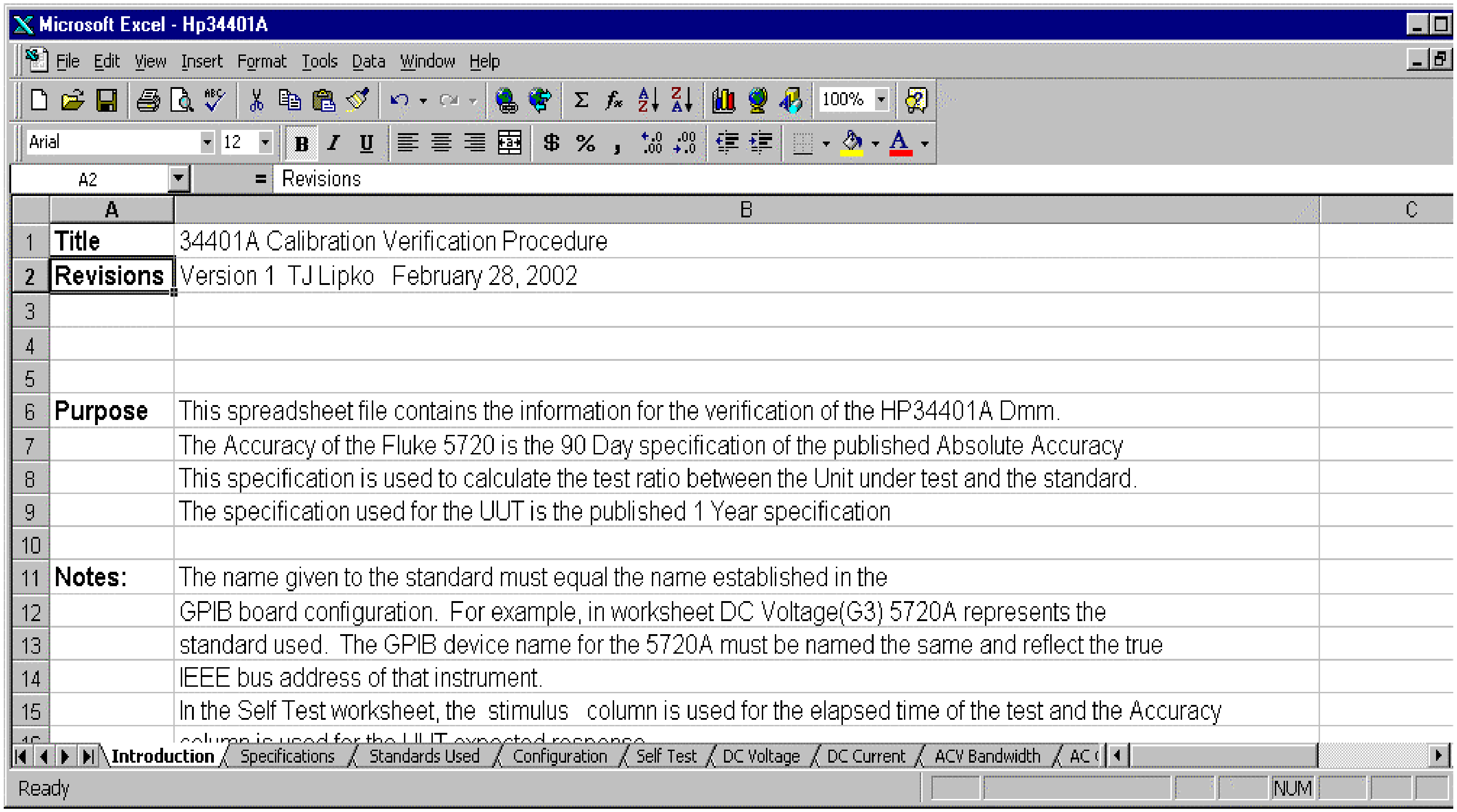


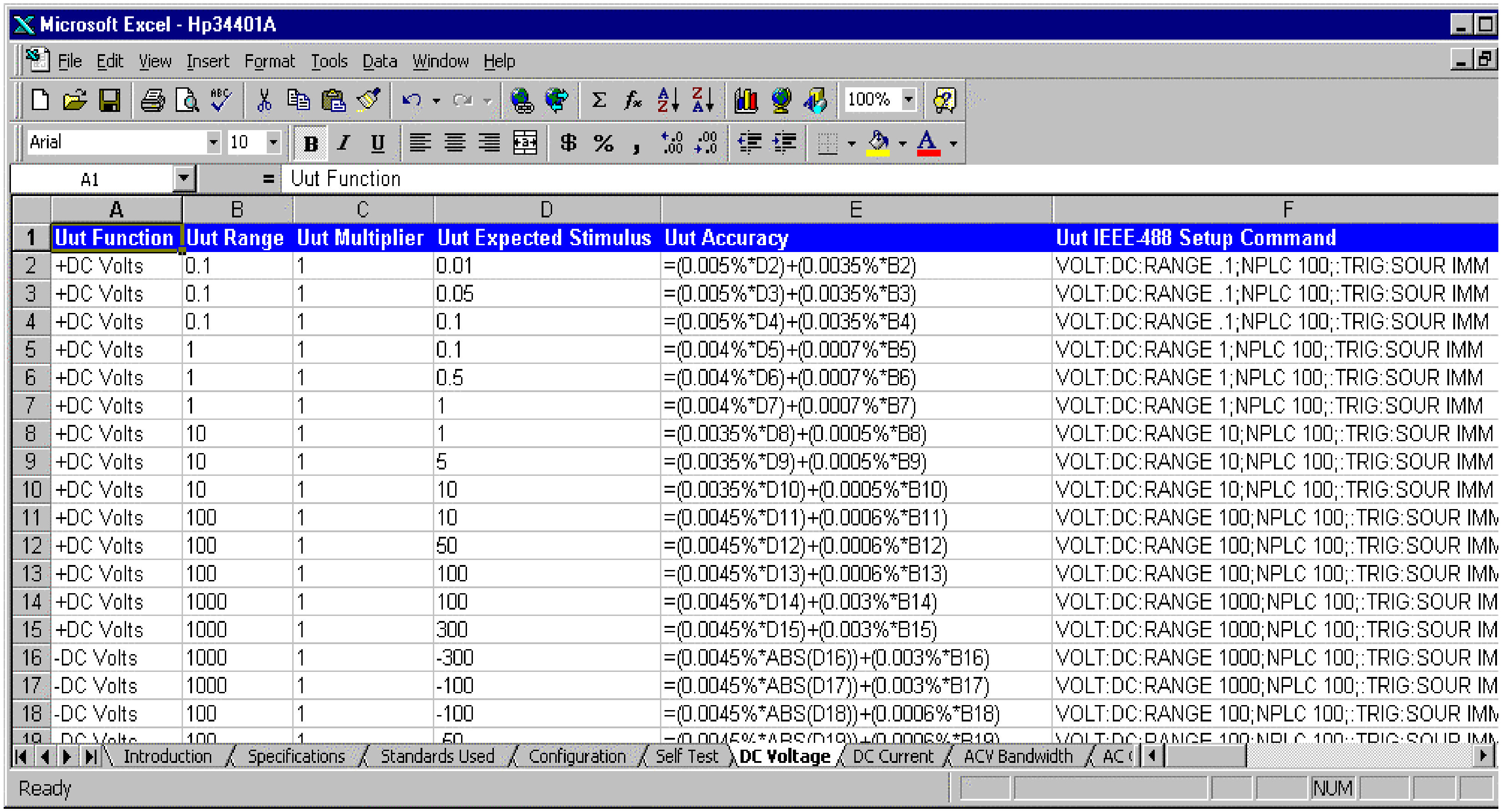
Figure 4- Summary

Test File Generator

The test file generator requires specific data entry in to the first 26 horizontal cells (A-Z) of the spreadsheet. This represent the test aspect, ranges, stimuli, communications, accuracy, etc. It may sound like a lot but really is not difficult to complete. When the first is completed then this can be used to cut and past to other files. Figure 5 shows an example of the test file. The first workbook is the title/revision page. Along with this is test definition and encountered problems of the instrument being tested.



All other workbooks contain a specific function or aspect of the UUT. Figure 6 shows the test of the DC function on the Agilent 34401A DMM.



	A	B	C	D	E	F
1	Uut Function	Uut Range	Uut Multiplier	Uut Expected Stimulus	Uut Accuracy	Uut IEEE-488 Setup Command
2	+DC Volts	0.1	1	0.01	=(0.005%*D2)+(0.0035%*B2)	VOLT:DC:RANGE .1;NPLC 100;:TRIG:SOUR IMM
3	+DC Volts	0.1	1	0.05	=(0.005%*D3)+(0.0035%*B3)	VOLT:DC:RANGE .1;NPLC 100;:TRIG:SOUR IMM
4	+DC Volts	0.1	1	0.1	=(0.005%*D4)+(0.0035%*B4)	VOLT:DC:RANGE .1;NPLC 100;:TRIG:SOUR IMM
5	+DC Volts	1	1	0.1	=(0.004%*D5)+(0.0007%*B5)	VOLT:DC:RANGE 1;NPLC 100;:TRIG:SOUR IMM
6	+DC Volts	1	1	0.5	=(0.004%*D6)+(0.0007%*B6)	VOLT:DC:RANGE 1;NPLC 100;:TRIG:SOUR IMM
7	+DC Volts	1	1	1	=(0.004%*D7)+(0.0007%*B7)	VOLT:DC:RANGE 1;NPLC 100;:TRIG:SOUR IMM
8	+DC Volts	10	1	1	=(0.0035%*D8)+(0.0005%*B8)	VOLT:DC:RANGE 10;NPLC 100;:TRIG:SOUR IMM
9	+DC Volts	10	1	5	=(0.0035%*D9)+(0.0005%*B9)	VOLT:DC:RANGE 10;NPLC 100;:TRIG:SOUR IMM
10	+DC Volts	10	1	10	=(0.0035%*D10)+(0.0005%*B10)	VOLT:DC:RANGE 10;NPLC 100;:TRIG:SOUR IMM
11	+DC Volts	100	1	10	=(0.0045%*D11)+(0.0006%*B11)	VOLT:DC:RANGE 100;NPLC 100;:TRIG:SOUR IMM
12	+DC Volts	100	1	50	=(0.0045%*D12)+(0.0006%*B12)	VOLT:DC:RANGE 100;NPLC 100;:TRIG:SOUR IMM
13	+DC Volts	100	1	100	=(0.0045%*D13)+(0.0006%*B13)	VOLT:DC:RANGE 100;NPLC 100;:TRIG:SOUR IMM
14	+DC Volts	1000	1	100	=(0.0045%*D14)+(0.003%*B14)	VOLT:DC:RANGE 1000;NPLC 100;:TRIG:SOUR IM
15	+DC Volts	1000	1	300	=(0.0045%*D15)+(0.003%*B15)	VOLT:DC:RANGE 1000;NPLC 100;:TRIG:SOUR IM
16	-DC Volts	1000	1	-300	=(0.0045%*ABS(D16))+(0.003%*B16)	VOLT:DC:RANGE 1000;NPLC 100;:TRIG:SOUR IM
17	-DC Volts	1000	1	-100	=(0.0045%*ABS(D17))+(0.003%*B17)	VOLT:DC:RANGE 1000;NPLC 100;:TRIG:SOUR IM
18	-DC Volts	100	1	-100	=(0.0045%*ABS(D18))+(0.0006%*B18)	VOLT:DC:RANGE 100;NPLC 100;:TRIG:SOUR IMM
19	-DC Volts	100	1	-50	=(0.0045%*ABS(D19))+(0.0006%*B19)	VOLT:DC:RANGE 100;NPLC 100;:TRIG:SOUR IMM

Figure 6

Conclusion

This application seems to have a place in calibration automation. We have been utilizing it now for about six months and it has been a very reliable tool for us. I would like to beta-test this application and receive feedback from colleagues to see if this method of automation is worth pursuing