

# Visual Basic for Applications with the Series Network Analyzers

Avtar Singh

Research Scholar, Manav Bharti University, Himachal Pradesh, India

## 1. INTRODUCTION

The Agilent Series of network analyzers with the built-in Microsoft Visual Basic for Applications (VBA) improves test automation dramatically. The built-in VBA facilitates the efficient development of a custom user interface for easy-to-use, intuitive analyzer operation in various measurement situations. High-speed data analysis, performed by the built-in VBA and without transferring data to an external PC, improves test throughput.

Finally, the integrated development environment of the VBA, including the powerful editor, significantly reduces development time. This product note explains how the built-in VBA can improve your network measurement environment by discussing the following points:

1. the benefits of the built-in VBA
2. examples of the effective use of the built-in VBA
3. the advantages of the powerful VBA editor
4. programming tips for network analyzer automation

## 2. Benefits of the built-in VBA

### Increases test engineer and operator productivity by automating testing

The built-in VBA dramatically improves test engineer and operator productivity. Since the Series contains an advanced programming capability, users can perform complicated repetitive tasks easily and without using the external PC, expediting the time it takes to obtain measurements. Automated measurements also decrease operator errors and improve operator productivity.

### Enables the customization of a flexible graphical user interface (GUI) for easier operation

The Series supports the customization of a flexible user interface through the use of powerful VBA GUI development tools. By combining a variety of GUIs with a mouse or the touch screen operation of the , a user can operate the analyzer easily and intuitively. Easy-to-use GUIs minimize operator errors and increase operator productivity.

### Enables high-speed post measurement data processing

With the ability to perform high-speed data transfers, the built-in VBA reduces post measurement data processing time. (Refer to figure 1.) To achieve superior data transfer speeds, the built-in VBA employs a common object model (COM) software interface.

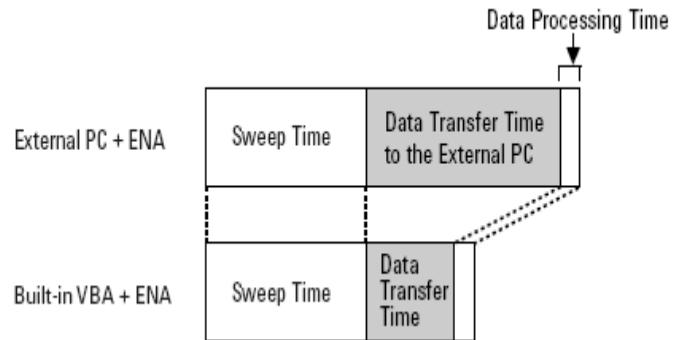


Figure 1. High-speed data processing by reducing data transfer time

### Reduces test system development significantly by using powerful editing functions

The built-in VBA helps developers save development time, reduce risks, and leverage their programming code. Easy-to-learn and easy-to-use, the VBA editor (Refer to figure 2) makes it simple to quickly design a GUI, write a program, and debug code.

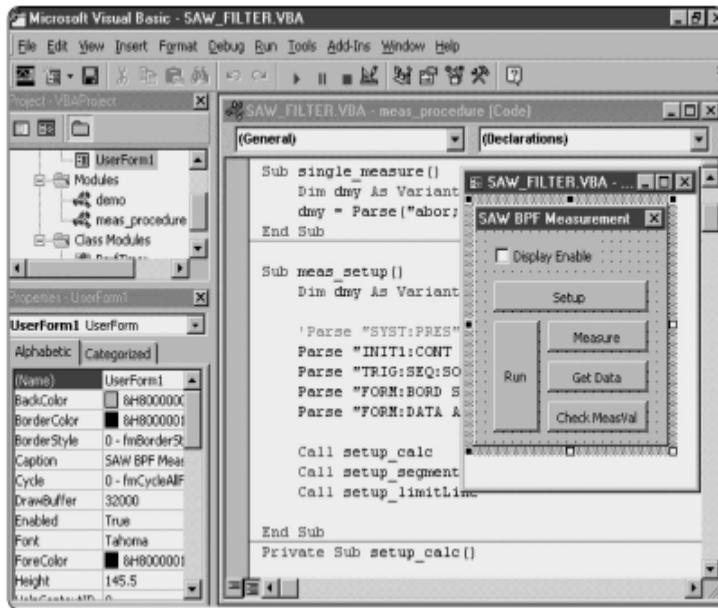


Figure 2. Built-in VBA editor

### 3. EXAMPLES OF THE EFFECTIVE USE OF THE BUILT-IN VBA

#### Improve test productivity by using automation

The built-in VBA can automate complicated repetitive operations and its GUI further simplifies the analyzer's operation. Figure 3 illustrates the use of the built-in VBA for conducting a surface acoustic wave (SAW) band pass filter pass/fail test. A series of measurement processes can be repeated automatically with a single click of the control button on the's GUI display. Furthermore, the program performs the limit test and displays the pass/fail result with key measurement parameters according to your needs.

Thus, the built-in VBA improves test productivity when certain levels of manual operation are required.



Figure 3. SAW bandpass filter pass/fail test

Another feature of the built-in VBA in test automation is its ability to control the as well as external instruments. The has a GPIB controller as standard equipment, which can control peripherals via the GPIB using a VISA library (figure 4). This function eliminates the external PC in an automated test system, simplifying the test system configuration and saving space.

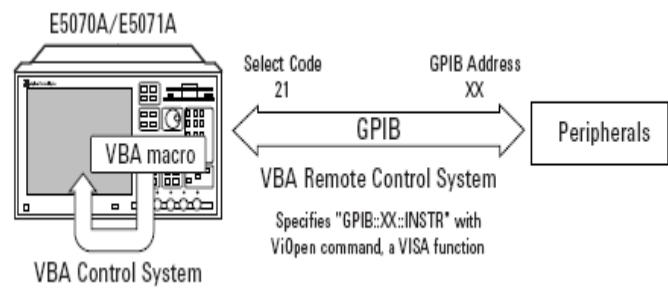


Figure 4. Example of configuring a control system using the built-in VBA

#### Customize a flexible GUI for efficient operation

The built-in VBA allows the customization of a user interface in order to accommodate specific measurement needs. Figure 5 shows the example of the ECal wizard program takes advantage of VBA GUI. By combining a rich GUI with a mouse or the touch screen operation of the , a user can operate the analyzer easily and intuitively. The ECal wizard program displays a graphical instruction, which increases operator productivity and minimizes operator errors. The GUI expands the 's capabilities and

allows you to create a test environment tailored to your needs.

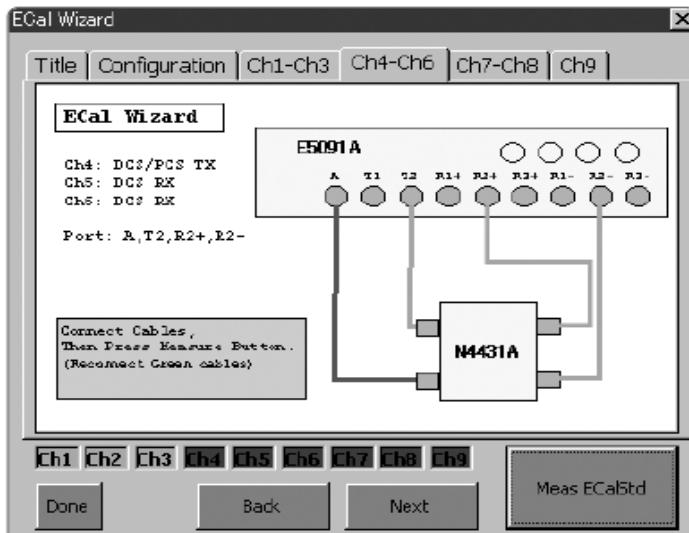


Figure 5. ECal wizard program to aid multiport calibration

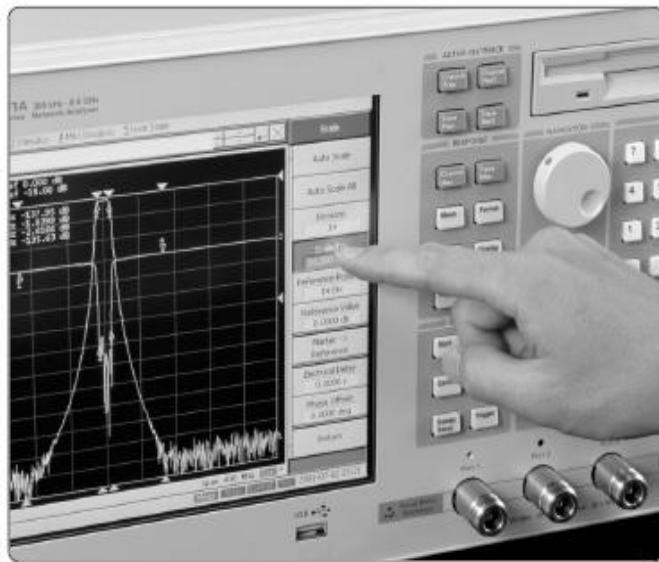


Figure 6. Touch screen eases interactive operation

### High-speed post measurement data processing

By eliminating the data transfer between an external PC and the , the built-in VBA reduces post measurement data processing time. This is very beneficial to the production

line, where post measurement data processing is required and yet high throughput needs to be maintained.

This section shows the measurement time comparison between the automation with the built-in VBA and the automation with the external PC in the characteristic impedance ( $Z_c$ ) evaluation of a balanced cable.

As shown in formula 1, four parameters ( $S_{dd11}$ ,  $S_{dd22}$ ,  $S_{dd12}$ , and  $S_{dd21}$ ) are necessary in order to calculate the  $Z_c$ .

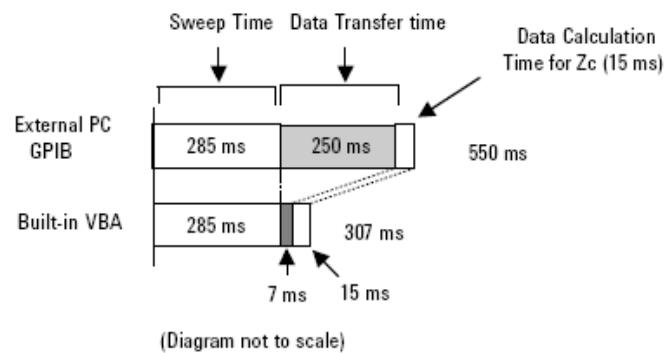
### Formula 1

$$Z_c = Z_0 \times \sqrt{\frac{(1 + S_{dd11})(1 + S_{dd22}) - S_{dd12}S_{dd21}}{(1 - S_{dd11})(1 - S_{dd22}) - S_{dd12}S_{dd21}} \frac{(1 + S_{dd11})(1 - S_{dd22}) + S_{dd12}S_{dd21}}{(1 - S_{dd11})(1 + S_{dd22}) + S_{dd12}S_{dd21}}}$$

All parameters are complex numbers.

Since the trace data of the four parameters has substantial size, data transfer time considerably effects the total cycle time. Figure 7 illustrates the cycle time from the beginning of the measurement to the end of  $Z_c$  calculation.

When the data analysis is performed on the external PC, the total cycle time substantially increases because of the data transfer time, while the built-in VBA minimizes data transfer time and improves test throughput. This is a realistic example of the benefits the built-in VBA offers high-speed production environments.



Conditions:	NOP: 801
Parameters:	$S_{dd11}$ , $S_{dd22}$ , $S_{dd12}$ , $S_{dd21}$
64 bit Binary Transfer	
Display Update OFF	

Figure 7. Improve throughput by reducing the data transfer time

At this point, it is worth discussing the handler interface of the , since the high-speed handshake with the parts handler is very important in the production lines requiring high-speed measurement. The handler interface enables fast output of the pass/fail result and the user defined I/O signal from the handler I/O port. Therefore, you can control the parts handler without compromising the high-speed data processing of the built-in VBA.

#### 4. EFFICIENT PROGRAM DEVELOPMENT USING VBA EDITOR

This chapter describes how to reduce the test development period by using the built-in VBA and is based on the program development flow shown in figure 8.

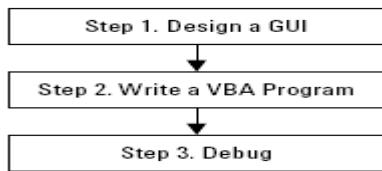


Figure 8. The flow of developing a VBA program

##### Step 1. Design a GUI

One of the most beneficial features of the VBA is that it is designed for easy GUI development, making it possible to design a wide variety of GUIs for intuitive equipment operation. This benefit is particularly significant when compared to conventional analyzer programming language, such as Instrument BASIC (IBASIC).

In addition to the standard VBA GUIs, the analyzer's softkeys can be easily customized in the VBA automation, as shown in figure 9. The combination of the VBA GUI and customized softkeys further enhances the analyzer's usability.

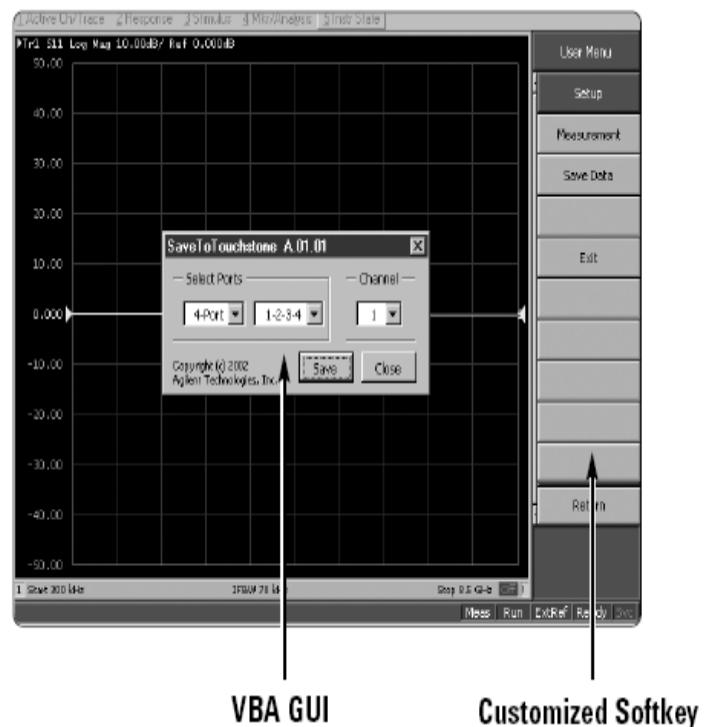


Figure 9. Soft key menu (the right side of the screen) and VBA's GUI

##### Step 2. Write a VBA program

The built-in VBA editor provides more powerful editing functions than conventional programming languages such as IBASIC. When you enter code, the editor displays lists of programming commands (including the control commands of the ) as shown in figure 10. Additionally, the editor indicates a typo by changing color (figure 11). These functions significantly expedite programming.

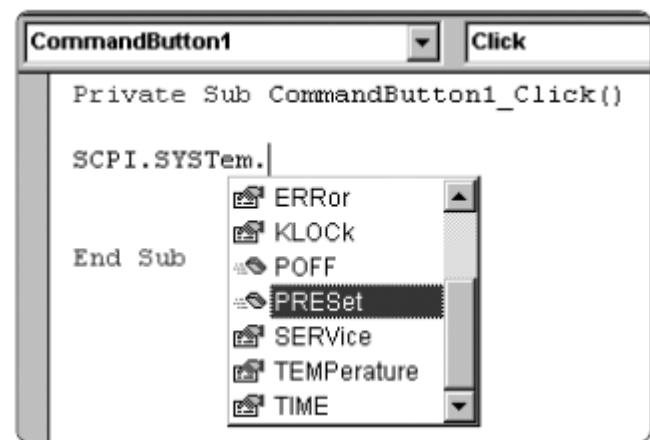


Figure 10. Auto list members feature

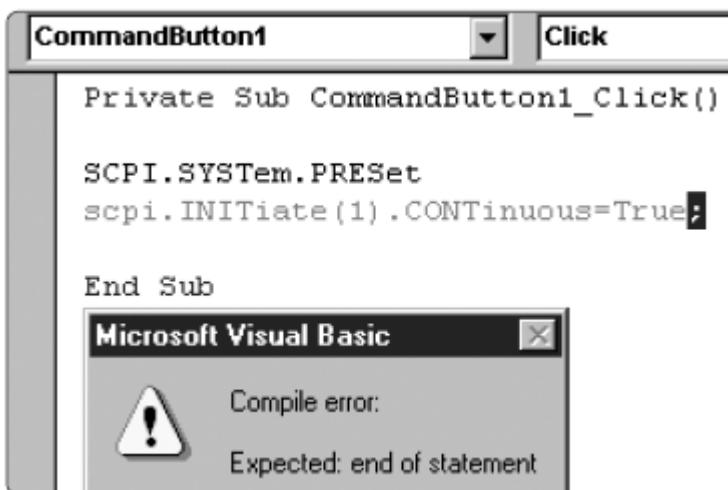


Figure 11. The built-in VBA editor indicates a typo

Since all Microsoft Office applications such as Word, Excel, and PowerPoint have VBA, code and illustrations can be developed in these applications and imported into VBA programs for the Series analyzers. Figure 12 illustrates how code developed for one project can be reused, shortening software development and allowing you to efficiently use your programming resources.

VBA program at a particular statement or monitoring the value of a data array easily can be done by placing a break point with a mouse click. When you point to the data array or formula, the VBA editor shows the current value. These functions help to quickly identify errors and reduce test development time.

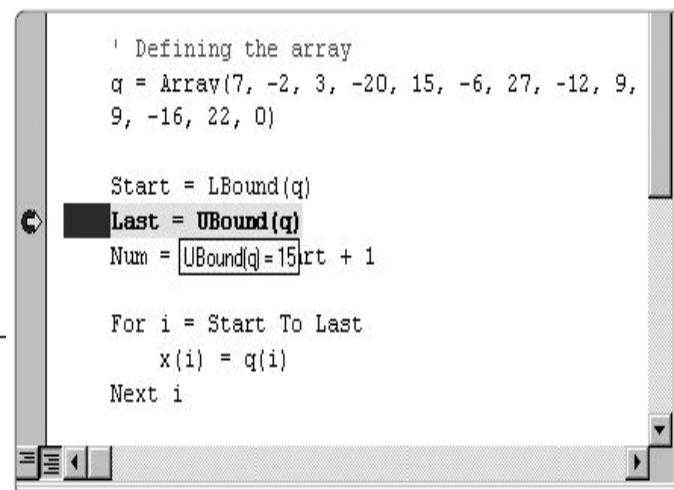


Figure 13. Setting a break point and monitoring a variable

## 5. PROGRAMMING TIPS FOR AUTOMATION

This chapter introduces programming tips for effectively using the built-in VBA.

### 1. Automatically loading a VBA program at power-on

The VBA program that satisfies the following conditions will be automatically loaded when the is turned on.

Program location : A:\ or D:\

Program file name : autoload.vba

### 2. Execute a program by pressing [MacroRun] key

A VBA program can be run with the [MacroRun] key if you insert a standard module "Module1" and write a subprogram "Main()" in the "Module1" as follows:

```
Sub Main ()
```

Write programming commands that you want to execute when [MacroRun] key is pressed

### Step 3. Debug

The built-in VBA editor provides a variety of debug tools that reduce the time required to fix errors. Suspending a

End Sub

### 3. Tips for improving test throughput

It is notable that the measurement time can be reduced considerably by turning off the display update. Thus, if measurement speed is very important, it is recommended that the display be updated only when necessary. The display update condition can be easily set with the following commands:

Commands related to the display update:

SCPI. Display. ENABLE Turns on/off the display update  
SCPI. Display. Update. Immediate Updates the display once when the update of the display is OFF

### 4. Developing a VBA program in an external PC

Microsoft Visual Basic, Microsoft Excel VBA, or Microsoft PowerPoint® VBA can be used to edit the VBA program for the . By using the following functions, you will be able to develop a VBA program efficiently on an external PC.

- **Use the import/export function for program transfer between the and a PC**

After writing a VBA program in Microsoft Excel or Microsoft PowerPoint, export each module separately. You then will be able to use them in the built-in VBA of the by importing each module.

- **Enable auto complete function in an external PC**

Even if you try to input the COM objects for the on the external editor, the auto complete function (figure 9) is not available. However, by referring the "E5070.tlb" library, the auto complete function becomes available on the external PC. (Note that it enables edit only. COM objects for the cannot be executed.)

The "E5070.tlb" can be download at [www.agilent.com/find/](http://www.agilent.com/find/)

- **Create pictures and write a VBA program within the Microsoft PowerPoint**

Pictures like a connection diagram are often useful for intuitive operation, as shown in the ECal wizard example (figure 5). If you use the Microsoft PowerPoint to create pictures, you can complete the majority of the application development within Microsoft PowerPoint, as it is equipped with a VBA editor.

### 5. Expand measurement capability using other applications' objects

The built-in VBA can refer to the library developed in Microsoft Visual Basic or Microsoft Visual C++®, which expands measurement capability.

The libraries that satisfy the following conditions are automatically referenced when the Series is turned on.

Library location : D:\Agilent

Extensions of libraries : olb, tlb, dll, or ocx

### 6. Waveform analysis using a built-in analysis library

The has the following Dynamic Link Libraries (DLL) in order to simplify data analysis:

- waveform analysis library
- complex operation library

The waveform analysis library is the most suitable for doing a ripple analysis in the filter evaluation, generating an easy ripple analysis without creating complicated analysis routines. The complex operation library is very useful when calculating complex numbers such as the Zc calculation (formula 1 on page 6).

### 6. CONCLUSION

The built-in VBA drastically improves your measurement environment in a wide variety of applications. The built-in VBA is a powerful tool for test automation, which enables efficient development of custom user interfaces and provides intuitive operation. For the manual operation environment, test automation with easy-to-use user interfaces improves operator productivity and minimizes operator error. For a production line that requires high-speed measurement, the high-speed post data processing of the built-in VBA substantially improves test throughput. Finally, the integrated development environment of the VBA dramatically decreases development time and reduces the development time and cost.