1603 St-Regis
D.D.O., Quebec H9B 3H7, Canada
Tel 514-684-4554 Fax 514-684-8581
E-mail: info@focus-microwaves.com
Website: http://www.focus-microwaves.com

Application Note 52

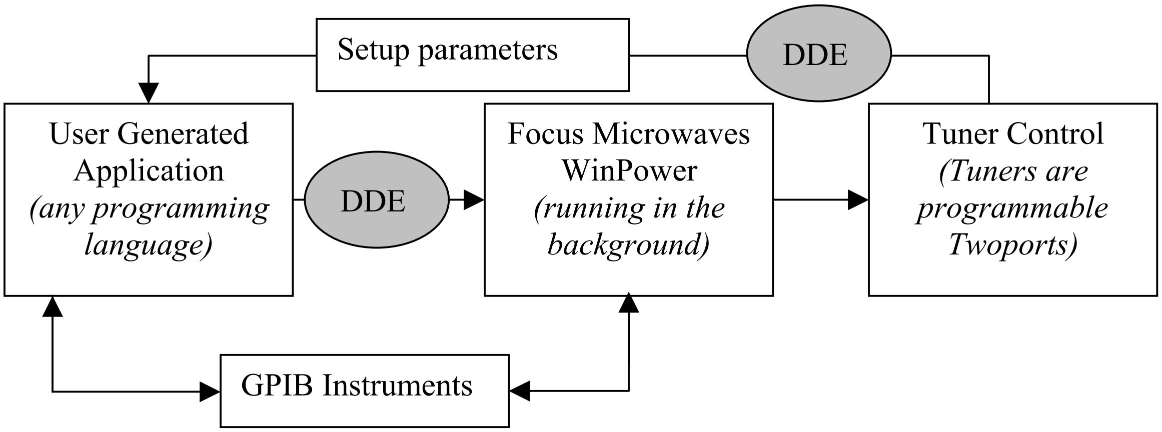
Tuner Control and Load Pull Using GPIB and Dynamic Data Exchange (DDE)

Summary

DDE operation allows customers to invoke and operate Focus' load pull measurement program WinPower from another application using Windows' capability of Direct Data Exchange. This other application can be either an existing commercial program or a User created program in any programming language, which supports DDE. Most functions of WinPower can executed by the other application written in LabView, HP-VEE, MatLab or Visual Basic. WinPower can also run macro-files sent by the other application [1].

Introduction

RF engineers definitely need the means to characterize RF devices at high power; Load Pull systems are thus becoming very important and useful tools. For certain applications, test engineers prefer to develop their own routines for measuring specific parameters. Focus' DDE capability offers an efficient approach for developing such applications. The load pull setup being defined in WinPower all measurements can be de-embedded to the DUT reference plane. Almost all major engineering test utility languages such as LabView, HP-VEE, MatLab and Visual Basic support DDE. Focus DDE allows easy synthesizing impedances using tuners and controlling GPIB instruments in order to design and execute User defined Load Pull test applications.


Setup parameters
DDE
User Generated Application (any programming language)
DDE
Focus Microwaves WinPower (running in the background)
Tuner Control (Tuners are programmable Twoports)
GPIB Instruments

June 2002

DDE Operation

WinPower uses the DDE (Dynamic Data Exchange) capability of Windows to exchange data and establish control between applications. Applications communicate with each other by establishing a ***DDE conversation***. The application that initiates the conversation is called the ***client***. The application that responds to the client's application is called the ***server***. When a client's application initiates a DDE conversation, it must identify the service name and the subject of conversation ***(topic).***

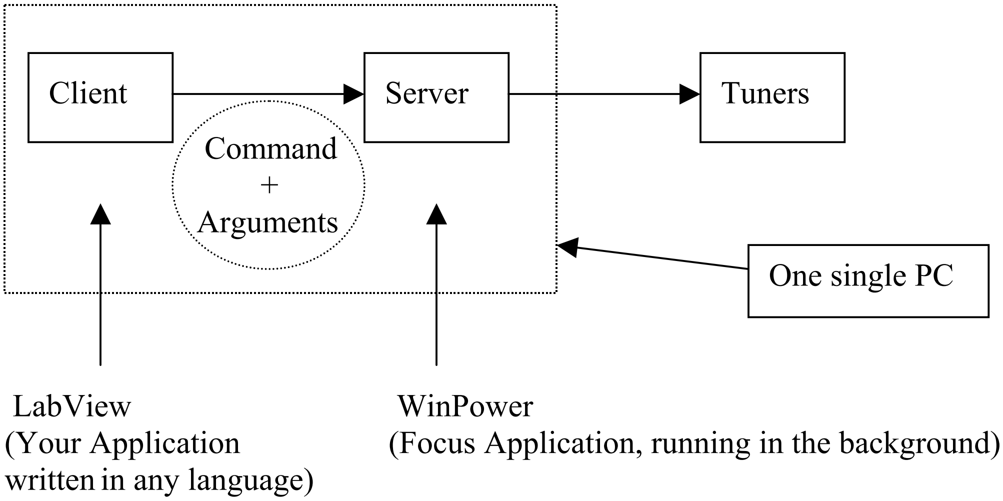


Figure 1: Memory operation of DDE

Description of Functions

There are two groups of functions, namely the ***Poke*** group and ***Request*** group. Poke commands are sent by external applications to WinPower and then WinPower will execute these commands.
For example, the command "Init 1" is POKED from your application, in this example made using LabView, WinPower will get this command, interpret and execute it. Tuner 1 is then initialized.
When the command "Loadpos" is REQUESTED from LabView to WinPower, WinPower will get the command, interpret and execute. Then the position of tuner 1 is returned from WinPower to LabView.

Table 1 gives the list of commands you can send to WinPower, which controls the tuners, to be executed. All arguments are sent in string format.

	Arguments	**Example of Argument**	**Descriptions**
Init	"tuner1 or 2"	"1"	Initialize the tuner 1
Inipht	"pht 1 or 2"	"1"	Initialize the harmonic tuner 1
Loadsetup	"setup file"	"focus.set"	Load the WinPower setup file
Tuneload	"mag phase(deg)"	"0.5 180"	Set the output tuner at the given Gamma Load
Tunesource	"mag phase(deg)"	"0.5 180"	Set the input tuner at the given Gamma Source
Sgpib	"add command"	"23 Volt?"	Send a GPIB command to an instrument
Rgpib	"add"	"23"	Read a GPIB instrument
Freq	"freq(GHz) "	"0.9"	Set the WinPower setup frequency
Sourcepos	"Xpos Ypos"	"1000 2000"	Move the source tuner at the X and Y pos
Loadpos	"Xpos Ypos"	"1000 2000"	Move the load tuner at the X and Y pos
Meas			Execute the WinPower measurement list from WinPower setup file
Loadcal	“calibration file”	“t1200200.cal”	Load calibration file and show tuner windows
Motorspeed	“data”	“300”	Change tuner moving speed
Pht1	“2fo 3fo”	“1000 1000”	Change harmonic tuner 1 position
pht2pos	“2fo 3fo”	“2300 2100”	Change harmonic tuner 2 position
tunepht1	“harmonic section, gamma, phase, regulation”	“2 1 200 0”	Tune harmonic tuner 2fo to nearest position of gamma=1, phase=200 degrees, Do not back-tune Γ(fo).
tunepht2	“harmonic section, gamma, phase, regulation”	“2 1 200 1”	Tune harmonic tuner 2fo to nearest position of gamma=1, phase=200 degrees, Do Γ(fo) back-tuning
Demo	“true or false”	“1”	Set WinPower on Demo mode
Ref	1—dut, 2—fixture 3—tuner	“1”	Set dut reference plane
caldir	“directory”	“c:\focus”	Set calibration file directory to c:\focus
axis	"tuner1 or 2 “	"2 x 2"	Switch tuner2 to 2fo (Combo tuner)
phtaxis	"pht 1 or 2"	"2 x 2"	Switch pht2 to 2fo
measlist	"file name"	"meas1.mls"	Load the WinPower measurement list of parameters meas1.mls
pindut	“power at pin dbm”	"25"	Set the pin 25dbm at DUT ref. plane
macro	"macro file"	"c:\focus\data\new.mac"	Execute the macro file new.mac

Table 1: Poke DDE commands and arguments acceptable by WinPower

Item	Returns	Example	Descriptions
Tuneload	"gamma, phase, loss"	"-0.5 0.0 1.2"	Gamma Load tuned and Output Power loss
Tunesource	"gamma, phase, loss"	"-0.5 0.0 1.3"	Gamma Source tuned and Input Available loss
Rgpib	"data"	"-10.2"	Read a GPIB instrument
FREQ	"freq(GHz)"	"1.2"	Selected Frequency inside the WinPower setup
Loadspar	"[Spar(Mag, Phase)]"	"0.5 180 .96 125 .96 125 .56 -30"	Tuner load S parameters, in mag and phase (deg)
Sourcespar	"[Spar(Mag, Phase)]"	"0.45 180 .95 145 .95 145 .46 –10"	Tuner source S parameters, in mag and phase (deg)
Sourcepos	"Xpos Ypos"	"1000 2000"	Tuner source X and Y pos
Loadpos	"Xpos Ypos"	"1000 2000"	Tuner load X and Y pos
Loadsetup	“file name”	“c:\focus\focus.set”	Get setup file name
tunephtsource	“mag phase”	“0.8 200 0.7 23”	Tune pht and get magnitude and phase of 2fo 3fo
tunephtload	“mag phase”	“0.8 200 0.7 23”	Tune pht and get magnitude and phase of 2fo 3fo

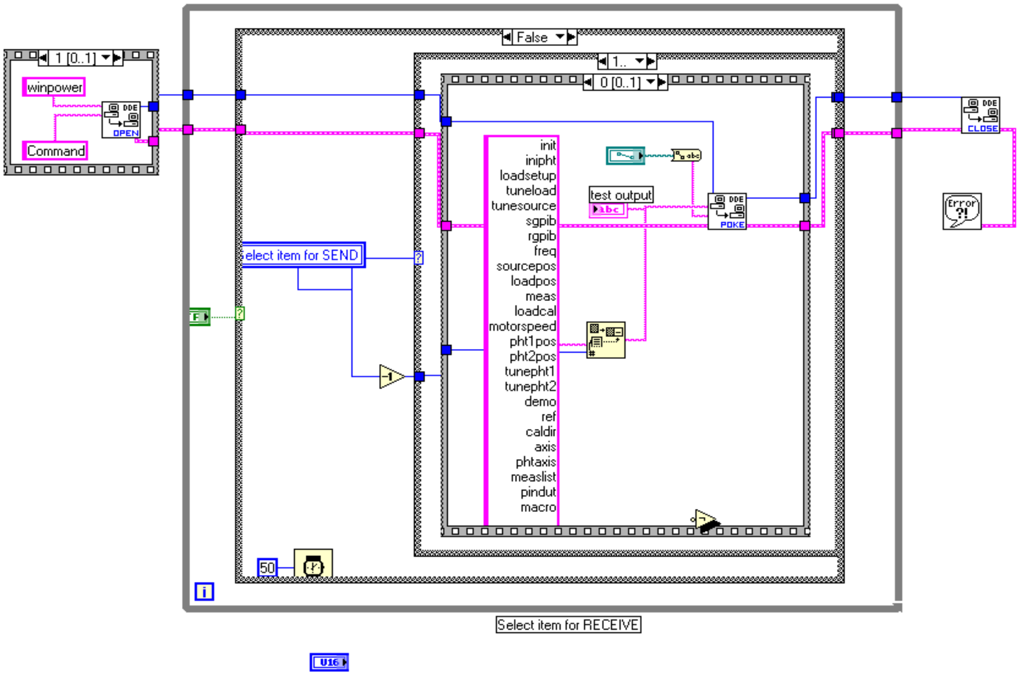
Table 2: List of request DDE commands allowing your application to retrieve data from WinPower. All the returned data is returned in string format.

Example in LabView programming language

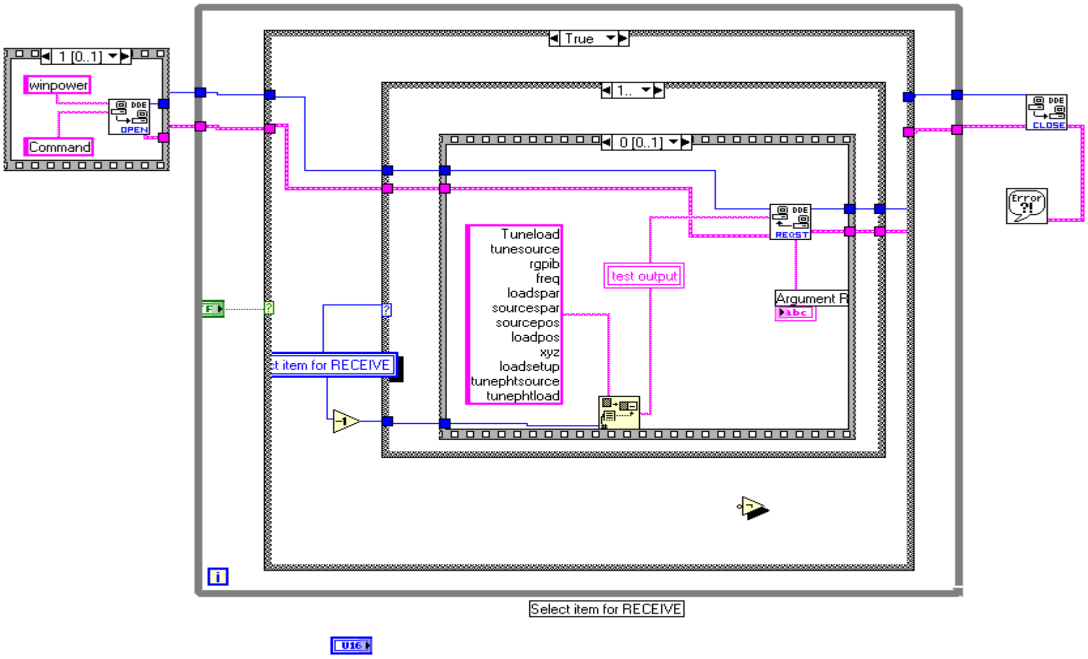
Following is an example programmed in LabView explaining how to control a tuner using WinPower.

All commands can be sent to WinPower to control the tuners, do the measurement, as well as retrieve data. You may make a load pull measurement easily by calling WinPower or make your own load pull measurement to tune tuners to certain impedance at different reference plane while acquiring data from instruments. It is your option to send GPIB commands directly from LabView (or any other language) to control and retrieve data and use WinPower only to control the tuners, change the reference plane of the measurement and retrieve S-parameters of the tuner twoport for de-embedding and setup corrections.

Sending Command



Receiving Command



Figures 2,3: Tuner control application programmed using LabView

Example in MATLAB programming language

Focus DDE control using MATLAB

```
%example for WinPower DDE communication
%this short program will initialize both tuners and then tune
%the load tuner to an interpolated S11 of 0.8, 180 degrees
%then it will move the same tuner to another (physical) position

clear all;

%initialize DDE from MatLab

channel=ddeinit('winpower','command');

%send command to init two tuners

rc=ddepoke(channel,'init',1);

rc=ddepoke(channel,'init',2);

%tuning load tuner to Gamma 0.8<180

rc=ddepoke(channel,'tuneload','0.8 180');

%Move tuner physical position to x=1000 y=2000

rc=ddepoke(channel,'LoadPos','1000 2000');

%for further details on MATLAB tuner operation see Focus AN 26
```

Example of Maury ATS control using GPIB commands

GPIB control of Maury ATS tuner

```
//Load GPIB DLL is done previously by the user

//send GPIB command to initialize tuner

ibwrt(add,”I”); //add = ATS controller GPIB address

//move one motor of tuner

ibwrt(add,”M2”);  //select motor

ibwrt(add,”F5”);//rate

ibwrt(add, ”s 225”);//slope

ibwrt (add, “R80”);//max rate

ibwrt(add,”Z 1”); //division
ibwrt(add,”+”); //direction

ibwrt(add,”7500);

ibwrt(add,”G”); //go
```

Conclusions

Focus tuner control can be used to generate User-defined operations, from simple tuner initialization and movements to complex load pull operations. This can be done using direct memory communication between applications, also known as DDE (Dynamic Data Exchange). Several test engineers use this capability to design their own applications using a variety of programming languages, such as LabView, HP-VEE, Visual Basic, MATLAB, C, C++, Pascal and other… These custom-made applications use tuner control via Focus software in parallel or exclusively with instrument GPIB control from these applications. Older versions of Focus tuner control using an external GPIB controller ETC (External Tuner Controller) have been entirely displaced by the modern and flexible DDE operation.

References

[1] “Macro File Operation of CCMT System”, Application Note 28, Focus Microwaves, November 1996.