

WILTRON

**Model 6700 Series
Swept Frequency Synthesizer
Instrument Driver
for LabWindows®**

User's Guide

Version 1.00

Locate the Serial Number decal sheet packaged with the manual, and paste the driver software serial number here.

Serial Number

You will be asked for this number when you call Wiltron Customer Service for support.

Wiltron

Important Notice

The material in this manual is subject to change without notice. WILTRON Company assumes no responsibility for errors that may appear in the manual. WILTRON Company makes no commitment to neither update nor maintain currency of the information contained in this manual.

Copyright

Under the copyright laws, this book may not be copied, photocopied, reproduced, translated — in whole or in part — without the written consent of WILTRON Company.

Copyright 1992 WILTRON Company

Trademarks

LabWindows® is a registered trademark of National Instruments Corporation

IBM is a registered trademark of International Business Machines Corporation. Personal System/2, IBM PC, PC AT, PC/XT, PC DOS, IBM CGA, IBM EGA, IBM VGA, and Micro Channel are trademarks of International Business Machines Corporation.

Microsoft, Microsoft QuickBASIC, Microsoft BASIC, Microsoft Visual BASIC, and Microsoft C are trademarks of Microsoft Corporation.

Preface

The 67XX LabWindows Instrument Driver User's Guide provides a tutorial and both general and detailed descriptions of the various functional panels displayed in the LabWindows environment. The user should be familiar with measurements using the applicable WILTRON instrument and with MS- or PC-DOS conventions. A knowledge of LabWindows, while helpful, is not essential. The WILTRON Instrument Drivers software can be used to create executable stand-alone application programs.

Manual Organization

The manual is divided into three sections:

Section 1, General, provides general information and a tree structure for the Instrument Driver functional panels.

Section 2, Using the 67XX Driver with LabWindows, provides description and a tutorial for using the driver within the LabWindows environment.

Section 3, Driver References, provides detailed descriptions of the functional panel and instrument controls. It also provides sample syntax and a listing of variable-type used in the program.

Table of Contents

Section 1 — General

Introduction	1-3
Requirements	1-3
Installing Instrument Drivers	1-4
Overview For LabWindows Users	1-4
Overview For Non-LabWindows Users	1-5

Section 2 — Using the 67XX Driver with LabWindows

Introduction	2-3
General	2-3
Loading 67XX Driver as Instrument Module	2-4
Debug Utility	2-6
Tutorial	2-8
Creating a Compiled Program	2-16

Section 3 — Driver References

Introduction	3-3
Functional Panel Structure	3-3
Functional Panel Descriptions	3-3
Alternate Sweep (fc.als)	3-6
CW Sweep (fc.cw)	3-8
Delta-F Sweep (fc.delta)	3-10
Special Sweep (fc.spec)	3-14
Start-Stop Sweep (fc.stst)	3-16
Measurement Functions (getmeas)	3-18
Initialize (init)	3-20
Set Frequency Definitions	
(init.freq)	3-22
Output Power Level (pc.level)	3-24
Output Power (pc.pwr)	3-26
Output Power Sweep (pc.pwrs)	3-28
Amplitude Modulation (setam)	3-30
Frequency Setup (setcw)	3-32
Frequency Modulation (setfm)	3-34
Marker Selection (setmk)	3-36
Pulse Modulation (setpm)	3-38
Recall Setup (sr.recall)	3-40
Save Setup (sr.save)	3-42
Utilities (ut)	3-44
Close (w6700.close)	3-46

Section 1

Introduction to 67XX Synthesizer LabWindows® Driver

Section 1

Introduction to 67XX Synthesizer LabWindows Driver

General

WILTRON Instrument Drivers software provides an easy-to-use tool for developing application programs for applicable microwave systems via the General Purpose Interface Bus (IEEE-488 Bus).

This software contains modules that automatically configure an applicable WILTRON instrument for use on the bus, along with high-level instrument control commands that save you the time required to learn and program the GPIB commands of the instrument.

Other benefits of the software:

- It automatically checks for proper bus functioning. If a command is sent to a bus instrument and no error is reported, the bus can be assumed to be working correctly.
- It makes the application program independent of connected WILTRON or other instruments. In other words, if one supported instrument is replaced by another, the program does not have to be modified; it only has to be reconfigured for the new instrument set-up.

Requirements

The WILTRON Instrument Driver software requires an IBM PC AT, PS/2, or compatible computer running MS- or PC-DOS, Version 3.0 or later.

The WILTRON Instrument Driver software requires National Instruments LabWindows version 2.2 or later.

The software is delivered on 5-1/4 inch 1.2 Mb Floppy disks and 3-1/2 inch 1.4 Mb floppy disks.

At least 2 MB of memory is required to run the LabWindows program — 4 Mb is recommended.

For Microsoft QuickBASIC, Professional BASIC, Visual BASIC, C, Quick C, and Borland C++ and Turbo C++ users, you can use the 67XX Instrument Driver software to produce compatible instrument-control-program code.

Installing Instrument Drivers

This section provides instructions for installing the WILTRON Instrument Driver. Proceed as follows:

- Insert the WILTRON driver diskette in your A: or B: drive, as appropriate.
- Change to the LabWindows, Instruments directory (*drive*\LW\INSTR), and type the following DOS command: COPY A: (B:) *.*. This copies the following four files to the target subdirectory: W6700.LBW, W6700.LWI, W6700.FP, W6700.DOC (All four of these files MUST reside within the same subdirectory.)
- Once the copying is completed, return the driver diskette to a safe storage location.

Overview For LabWindow Users

LabWindows is a software development system for BASIC, C, and C++ programs (see page 1-3 for listing of supported languages). It contains an interactive environment for developing programs with drivers and libraries (functions) for creating data acquisition and instrument control applications. LabWindows contains a comprehensive set of software tools for data analysis, data presentation, and high level instrument control.

The interactive program is an environment for editing and debugging BASIC and C (C++) programs. In the LabWindows environment, you can use the functions in the instrument drivers or libraries to write your program. In addition each function has an interface called a function panel that lets you interactively execute the function or generate code for calling the function.

The interactive program uses extended memory. Programs executed in the interactive program can use up to 16 megabytes of memory, depending on your computer configuration. Programs that run in the interactive program, however, must adhere to the LabWindows subsets for BASIC, C and C++.

Programs developed with the drivers and library functions can be run within the interactive program, or they can be compiled and linked into a stand-alone application (*.EXE) or run-time application (*.RTM) file. To help you create a stand-alone program, the LWMAKE utility automates the compile and link processes.

The real power of LabWindows lies in the libraries. They have functions for developing all phases of your data acquisition and instrument control system. For controlling the 67XX, Lab Windows has the Instrument Drivers Library. Programs that call this library can be developed with the interactive program. This program has tools that make program development quicker and easier.

LabWindows gives you the capability to execute instrument drivers with the aid of panels and thereby make programs in a very easy way. The panels contain items that can be selected to build and execute a driver. The drivers are separately declared in the Instrument Drivers Library.

Two advantages of using Lab Windows are:

- When writing an application program you do not have to remember all of the parameters that belong to the driver.
- Error reporting is shown automatically in the panels.

Overview for Non-LabWindows Users

Programmers who do not use LabWindows will also benefit from the WILTRON Instrument Driver software:

- You will not have to understand arcane GPIB codes to program applications for the 67XX Synthesizer. The driver software effectively manages low-level GPIB I/O operations and native instrument control.
- You will see greater program reliability because of the driver's extensive error-checking routines.
- You will see greater instrument independency, because instrument types and versions can be interchanged without having to reprogram the application.
- You will see reductions in the time required to develop, test, and debug applications.

Section 2

Using the 67XX Instrument Driver with LabWindows®

Section 2

Using the 67XX Instrument Driver with LabWindows

Introduction

This section provides an introduction to the LabWindows environment and a tutorial describing the use of the 67XX driver within LabWindows. This section assumes that you have read Part 1 of the National Instruments *Getting Started with LabWindows* manuals and are generally familiar with the Lab Windows screen and principles of navigation within the environment.

General

The following procedure describes how to access LabWindows and load files.

- Move to the directory containing the LabWindows executable (*.EXE) files. (This directory is usually named \LW.)
- Type LW.
This places you in the PROGRAM window of the LabWindows environment (below).



Loading 67XX Driver as Instrument Module

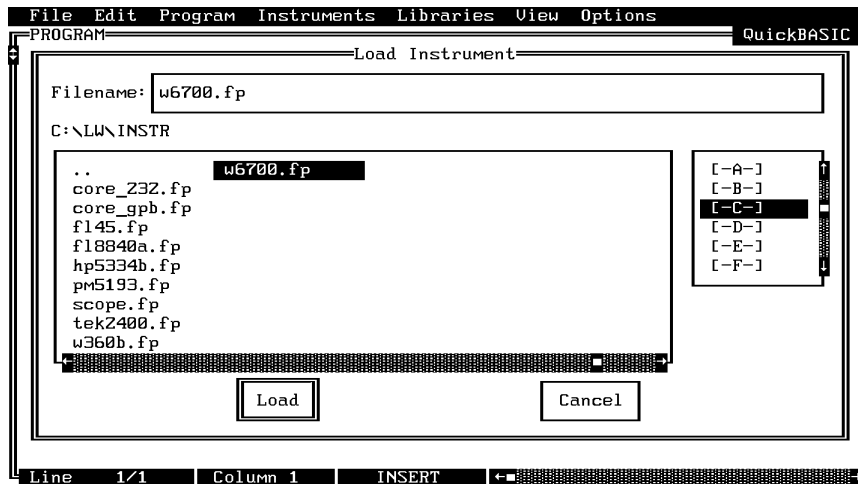
The following procedure describes how load the 67XX Driver as an instrument module.

- Place the cursor on **Instruments** in the top menu bar (below) and display the pull-down menu.



Select **Load**, then change to the d\LW\INSTR subdirectory (below).

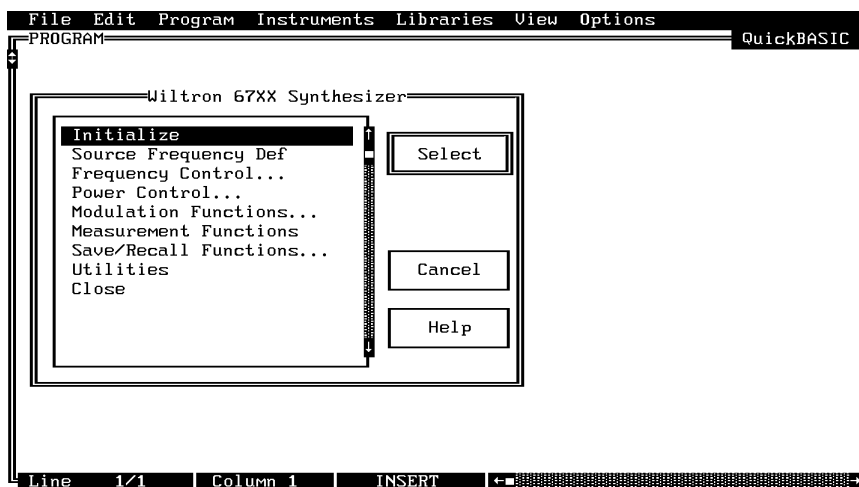
- Move cursor to **W6700.fp** and select **Load**.



- Move cursor to **Instruments** and display pull-down menu,.



- Select **Wiltron 67XX Synthesizer**, and observe that the 67XX main panel appears (below).

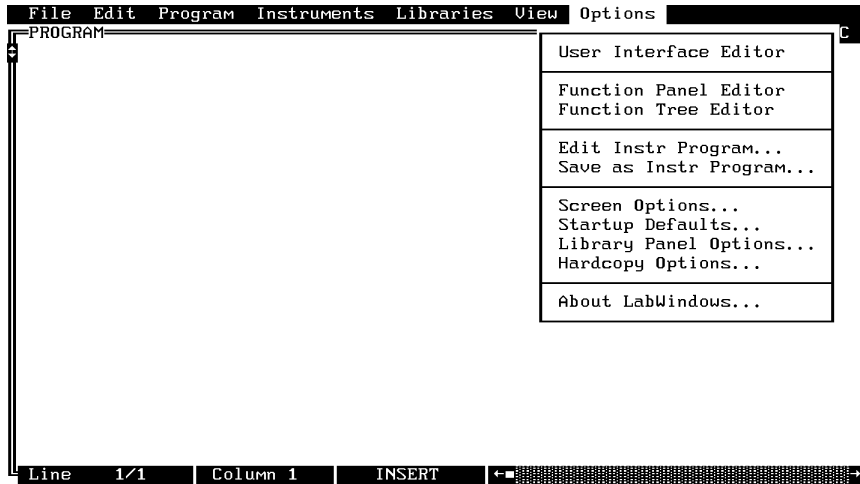


- You are now ready to proceed with developing control code using the 67XX Synthesizer instrument driver.

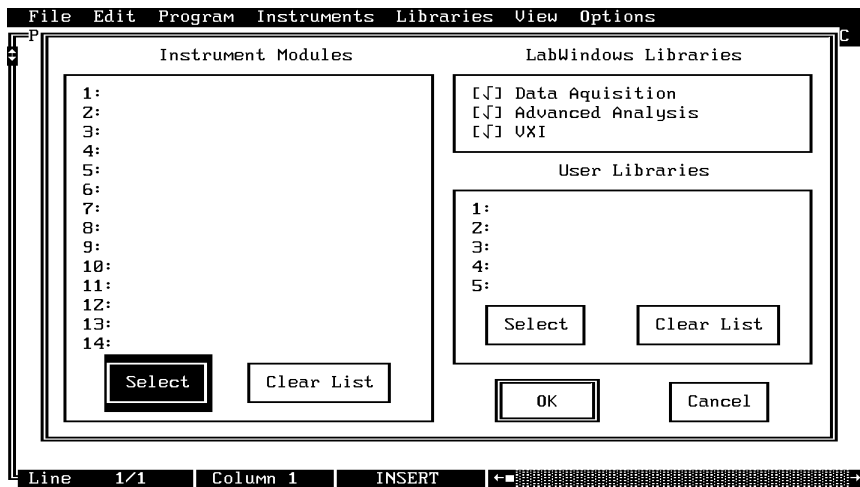
Loading 67XX Driver At Start-up

The 67XX Synthesizer driver can be automatically loaded each time LabWindows is started. The procedure for making this happen is given below.

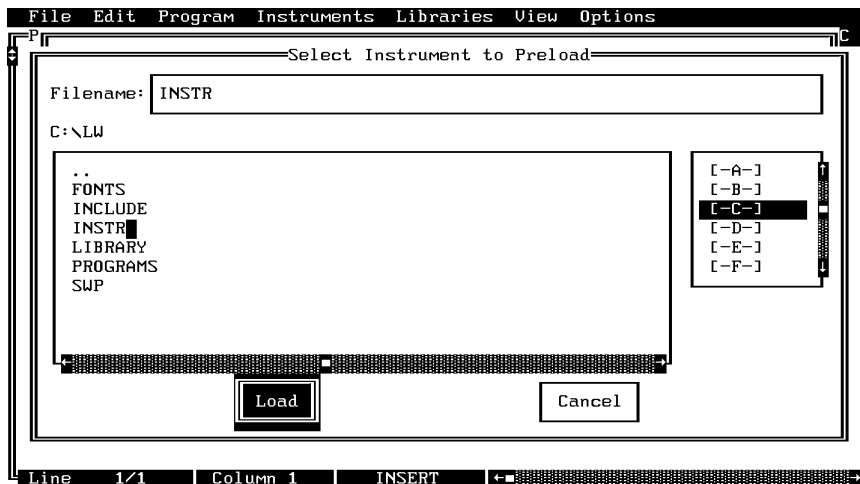
- Place yourself into the LabWindows PROGRAM window as was described on page 2-3.
- Move cursor to **Options**, on top menu bar, and display pull-down menu.



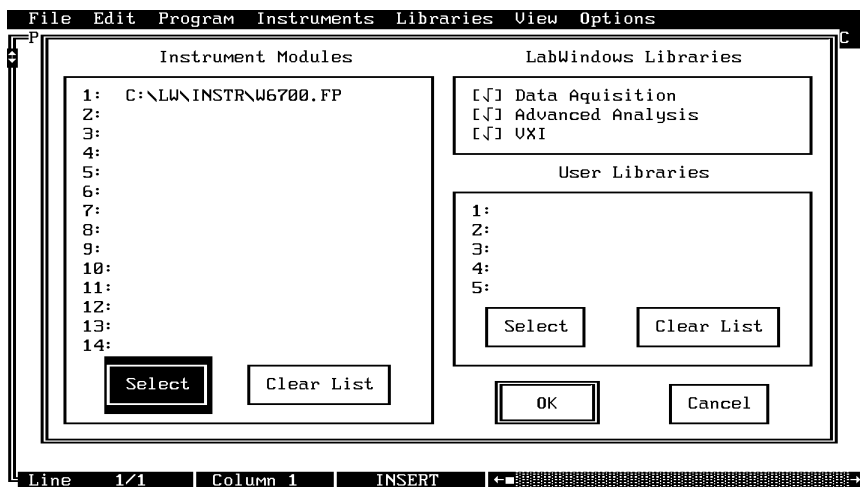
- Choose **Select** to locate the 67XX Synthesizer; alternatively, type in the path and file name.



- Choose the **INSTR** subdirectory from the file list, and click on **Load**.



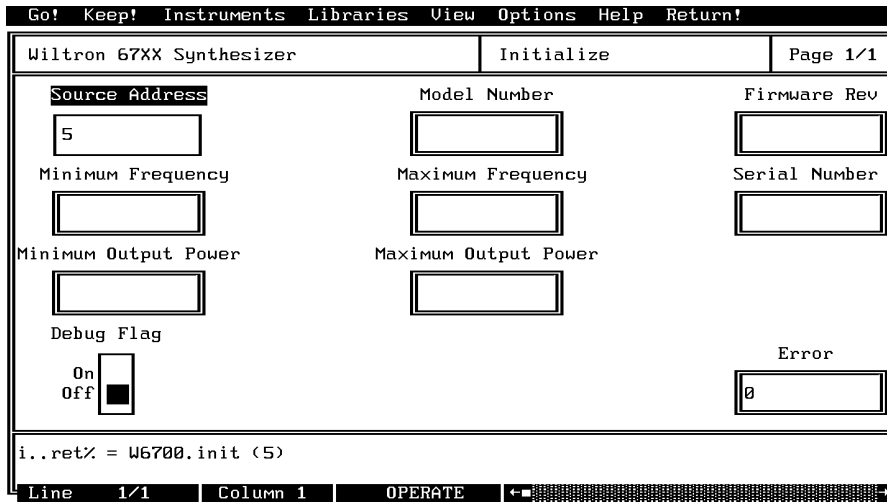
- Choose **W6700.fp** from the displayed file list, and click on **Load**.



Debug Utility

The 67XX Synthesizer LabWindows driver incorporates a unique Debug utility. This utility provides real-time error checking when the Debug Flag function is set to on within the program (debug%).

During program development: When the **Debug Flag** switch in the Initialization panel (below) is set to On and an error is detected, it returns one of the following error codes:



- 300 - No Response
- 301 - Parameter Range Error
- 305 - Unexpected SRQ.
- 309 - Modulation Range Error
- 311 - Self Test Failed

NOTE

Error message 301 is returned whether or not the debug switch is on. If the Debug utility is off and this message occurs, turning Debug on provides a specific error message (see above).

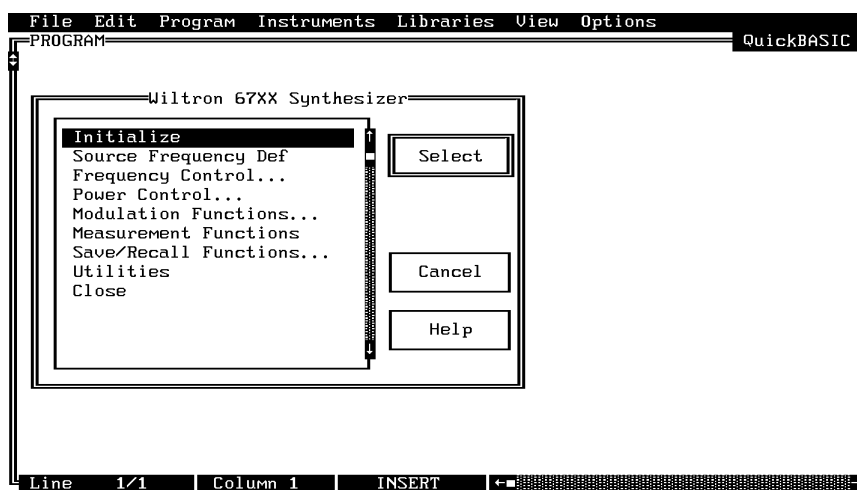
When the Debug Flag utility is off (DEBUG=0), *no error messages are returned* (except for 301).

When the function is set to on, execution of the program is slowed. Its use adds 300 ms to the execution of each command string. Consequently, we recommend that it be enabled only during program development, and that it be disabled before the program is compiled; that is, before you use the LWMAKE utility.

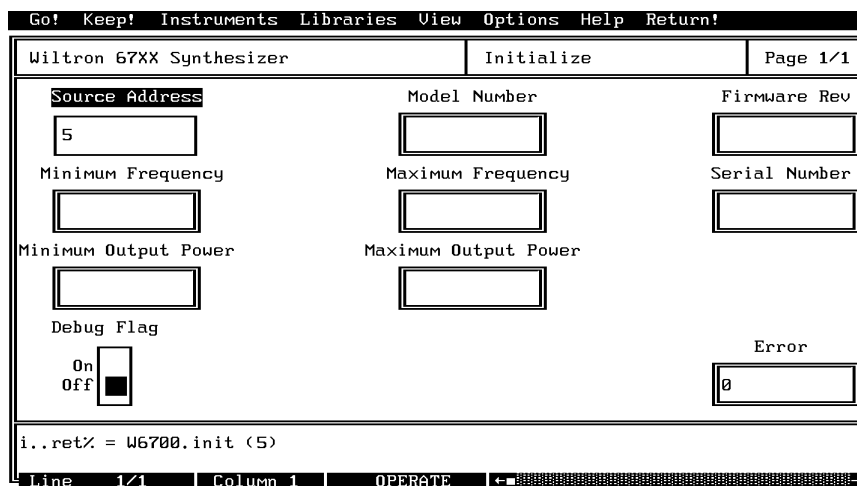
Tutorial

This tutorial takes you through the development of a simple program. This program initializes the 67XX synthesizer, assigns sweep frequencies and modulation parameters, and closes the driver. When you have finished stepping through the tutorial, you will have a BASIC program that can be compiled as a DOS executable (*.EXE) file. A Microsoft C program could be produced in the exact same manner by switching the native language to C (under the **Program** menu).

- Move cursor to **Instruments** in the top menu bar, and select **Wiltron 67XX Synthesizer**. This will display the pull-down menu shown below.



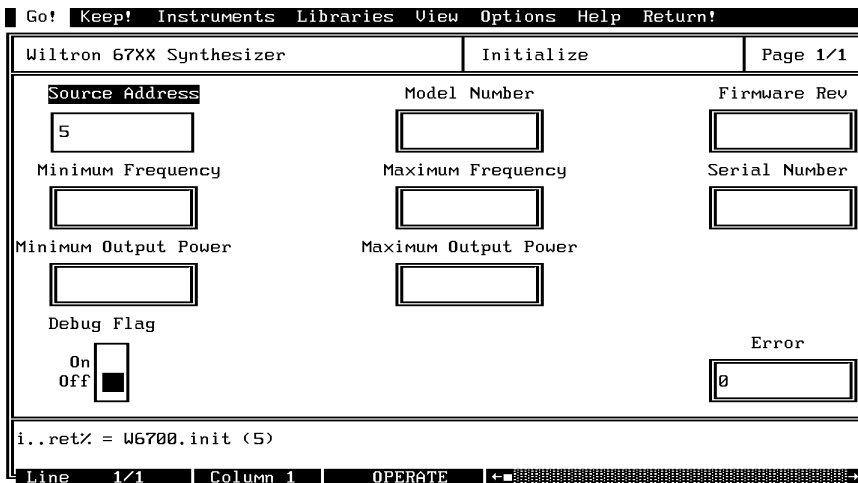
- Choose **Select** to display the Initialize panel (below).



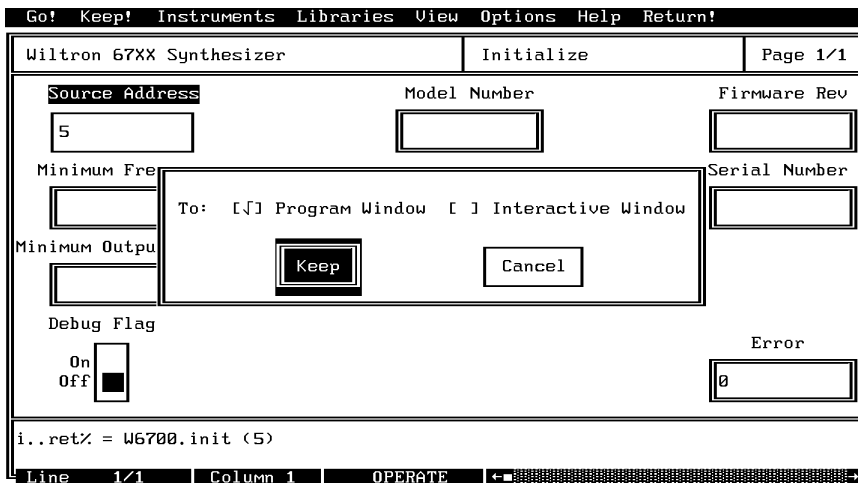
- With the **Source Address** highlighted as shown, enter the 67XX GPIB address (a number between 1 and 30; 5 is the default).

67XX LabWindows Drivers User's Guide

- Select **Go!** (below) in the top menu bar. If a 67XX is connected and set to GPIB address 5, it will respond by resetting itself and returning an identification string that will fill the **Model Number**, **Firmware Rev**, **Minimum Frequency**, **Maximum Frequency**, **Serial Number**, **Minimum Output Power**, and **Maximum Output Power** fields.
- Leave Debug Flag set to Off. This switch was discussed on page 2-6.

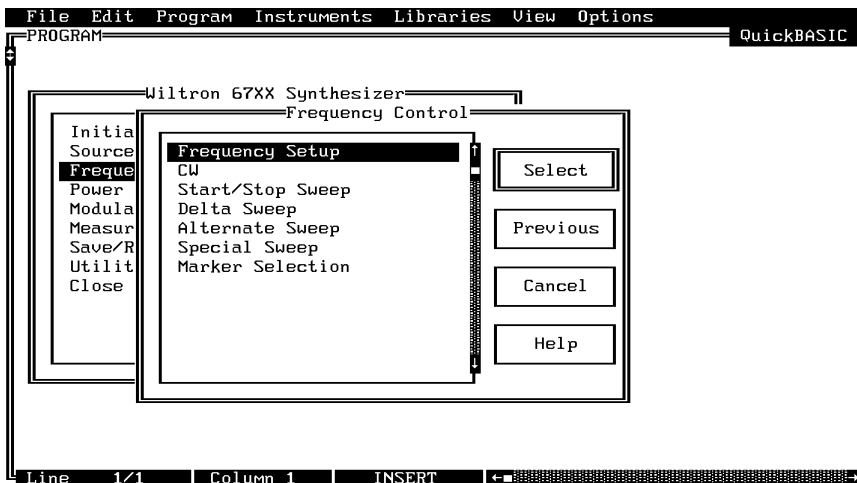


- Move the cursor to **Keep!**, in the top menu bar, then choose **Keep** from the next window (below) to select the default option. This transfers the code shown at the bottom of the panel to the PROGRAM window of the LabWindows environment.

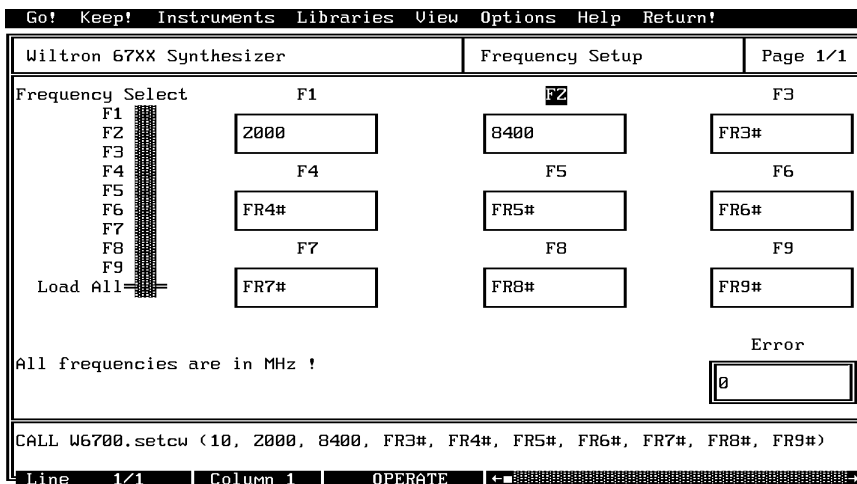


- Select **Instruments**, in the top menu bar, then **Wiltron 67XX Synthesizer** to return to the 67XX Driver main panel.

- Select **Frequency Control**, from the main menu, then **Frequency Setup** when the Frequency Control menu appears (below).



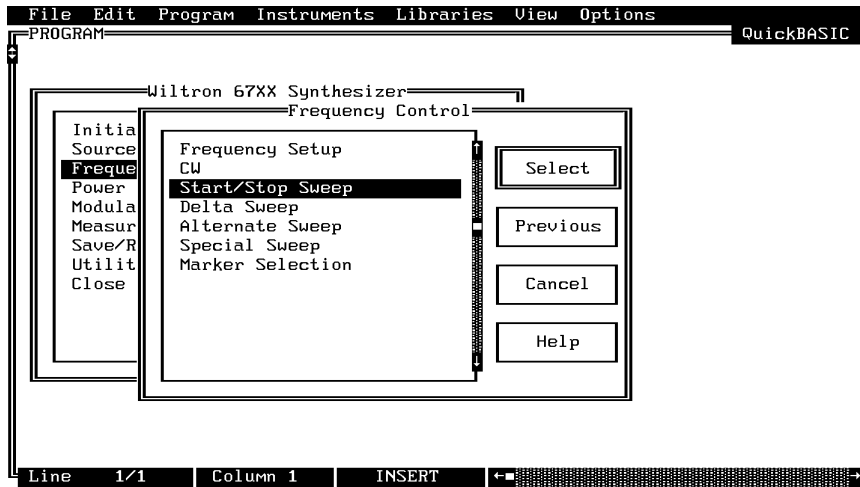
- Select **F1** (below), and enter **2000** from the keyboard.
- With **F2** highlighted, enter **8400** from the keyboard.
- Select **Load All**, from the **Frequency Select** control.



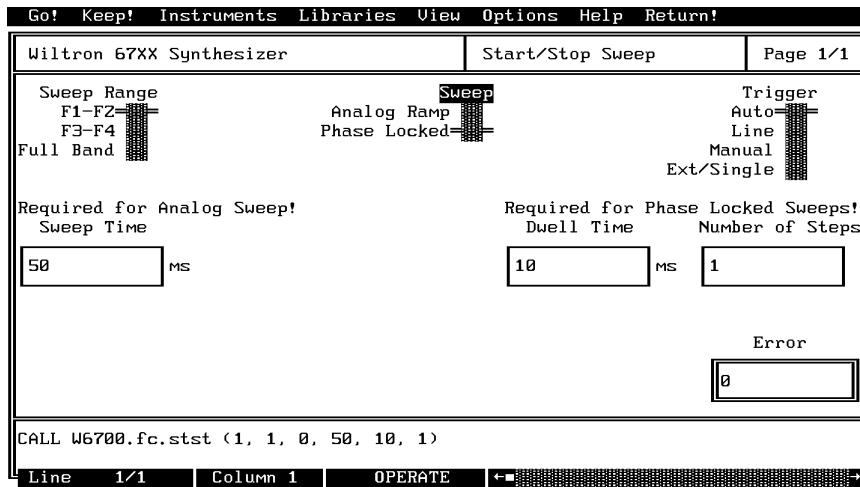
- Select **Keep!**, in the top menu bar; then choose **Keep** again when the next menu appears.
- Select **Instruments**, in the top menu bar, then **Wiltron 67XX Synthesizer** to return to the 67XX Driver main panel.

67XX LabWindows Drivers User's Guide

- Select **Frequency Control** and **Start/Stop Sweep** from the next menus to appear (below).

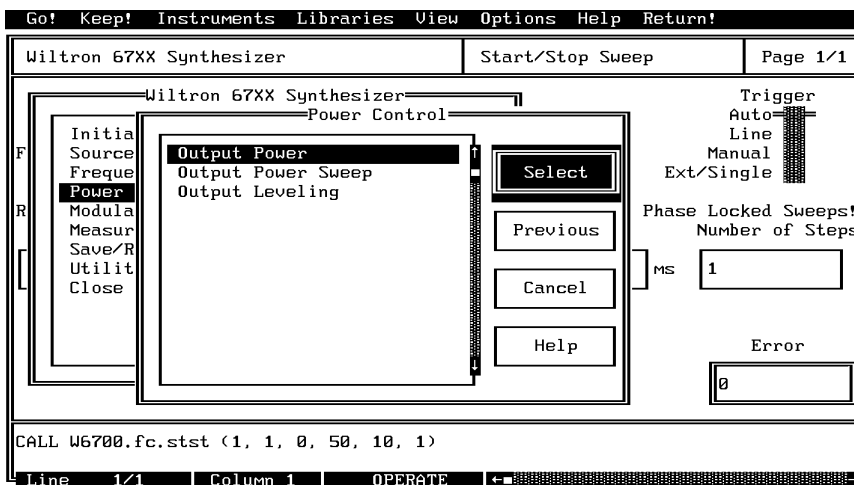


- Select **F1-F2** sweep range (below), **Phase Locked** sweep, **Auto** trigger, and the default settings of **50 ms** sweep time, **10 ms** dwell time, and **1** step.

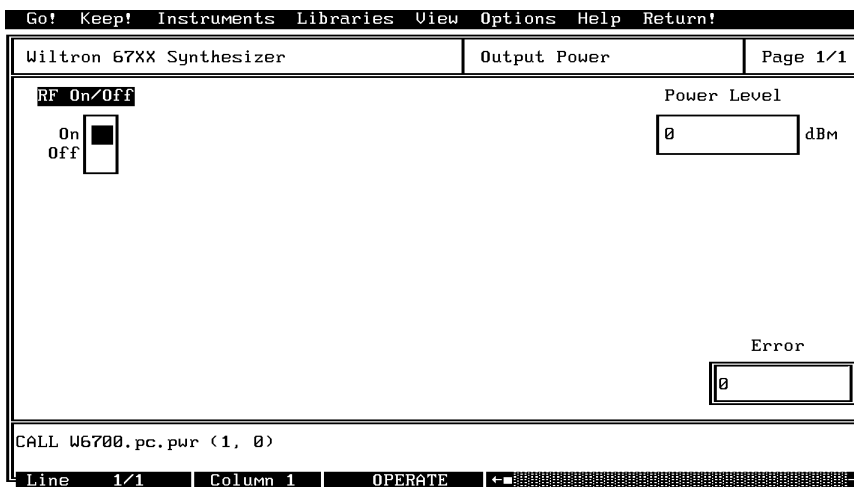


- Select **Keep!**, in the top menu bar; then choose **Keep** again to add the code to the program window.
- Select **Instruments**, in the top menu bar, then **Wiltron 67XX Synthesizer** to return to the main driver panel.

- Select **Power Control** then **Output Power** (below) to display the output power panel (next menu).



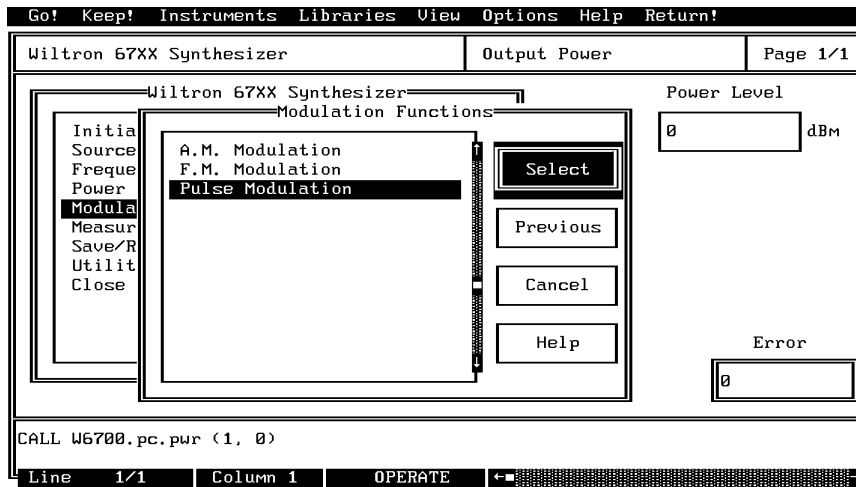
- Verify that the **RF On/Off** control (below) is **On**.
- Select **Power Level** and enter **-5** from the keyboard.



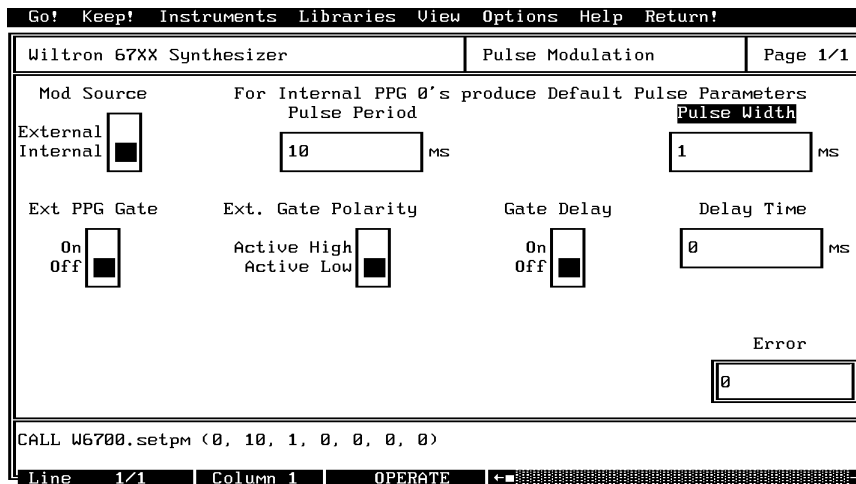
- Select **Keep!**, in the top menu bar; then choose **Keep** again to add the code to the program window.
- Select **Instruments**, in the top menu bar, then **Wiltron 67XX Synthesizer** to return to the main driver panel.

67XX LabWindows Drivers User's Guide

- Select **Modulation Functions** from the first menu, then **Pulse Modulation** from the second (below) to display the pulse modulation menu (below).

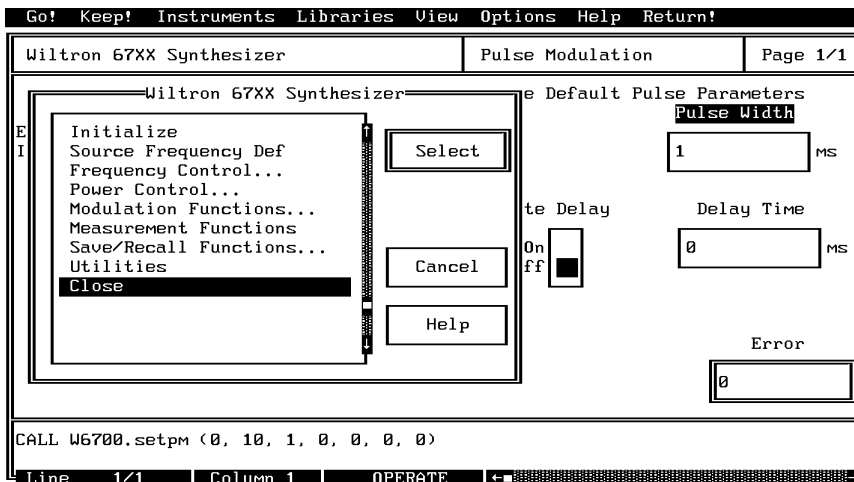


- Position **Mod Source** to **Internal** (below).
- Select **Pulse Period**, and enter **10** from the keyboard.
- Select **Pulse Width**, and enter **1** from the keyboard..
- Position **Ext PPG Gate** control to **Off**.
- Position **Ext Gate Polarity** control to **Active Low**.
- Position **Gate Delay** control to **Off**.

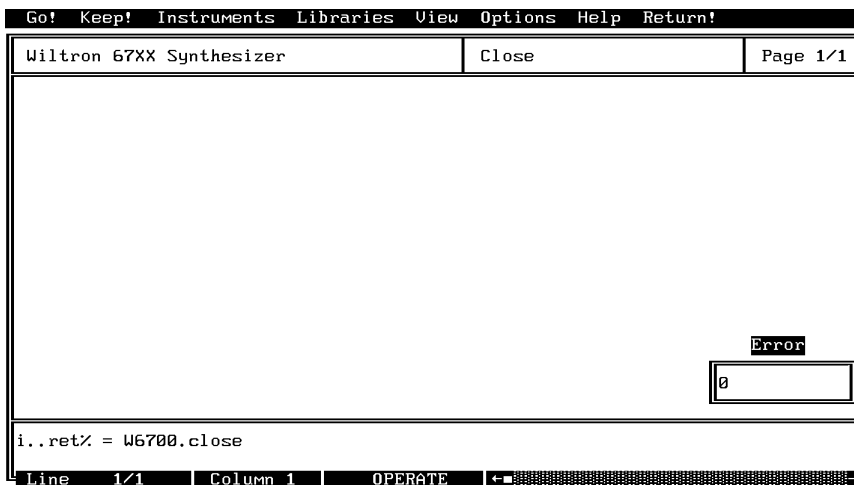


- Select **Keep!**, in the top menu bar; then choose **Keep** again to add the code to the program window.

- Select **Instruments**, in the top menu bar, then **Wiltron 67XX Synthesizer** to return to the main driver panel.
- Move cursor to **Close** (below) and choose **Select** to display the Close panel



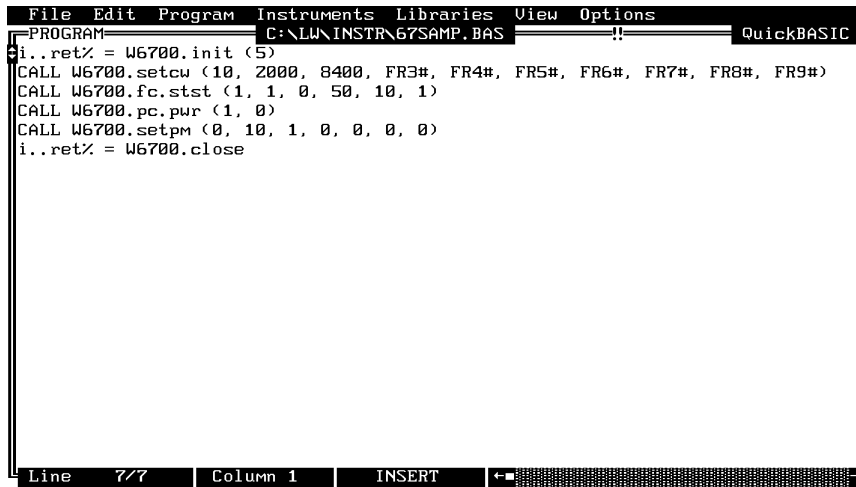
- There is nothing you need to do with this panel, except to select **Keep!**, in the top menu bar; then choose **Keep** again to add the code to the program window.



- Select **Return!** on the menu panel to return to the PROGRAM area.

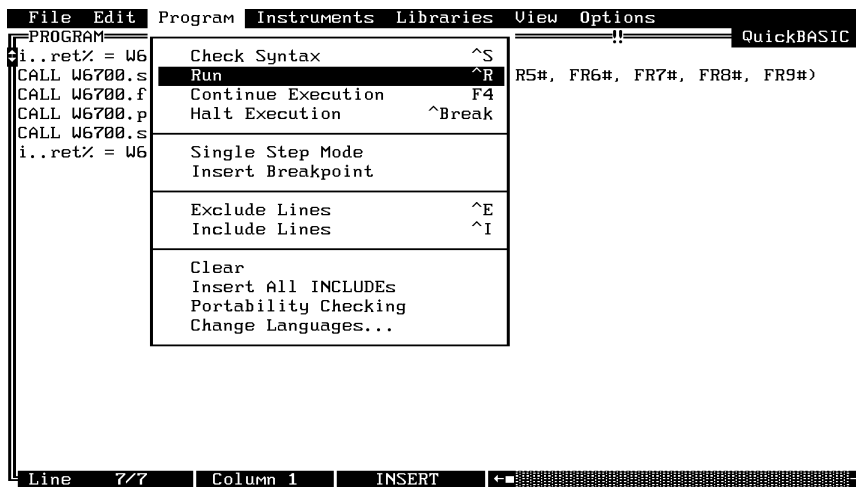
67XX LabWindows Drivers User's Guide

- The program that you have developed now appears (below).



```
File Edit Program Instruments Libraries View Options
PROGRAM C:\LW\INSTR\67SAMP.BAS QuickBASIC
i..ret% = W6700.init (5)
CALL W6700.setcw (10, 2000, 8400, FR3#, FR4#, FR5#, FR6#, FR7#, FR8#, FR9#)
CALL W6700.fc.stst (1, 1, 0, 50, 10, 1)
CALL W6700.pc.pwr (1, 0)
CALL W6700.setpm (0, 10, 1, 0, 0, 0, 0)
i..ret% = W6700.close
Line 7/7 Column 1 INSERT
```

- To test the program, move the cursor to **Program**, in the top menu bar and select **Run** (below) from the pull-down menu.



```
File Edit Program Instruments Libraries View Options
PROGRAM C:\LW\INSTR\67SAMP.BAS QuickBASIC
i..ret% = W6
CALL W6700.s
CALL W6700.f
CALL W6700.p
CALL W6700.s
i..ret% = W6
Check Syntax ^S
Run ^R
Continue Execution F4
Halt Execution ^Break
Single Step Mode
Insert Breakpoint
Exclude Lines ^E
Include Lines ^I
Clear
Insert All INCLUDEs
Portability Checking
Change Languages...
Line 7/7 Column 1 INSERT
```

- If there are no program errors triggering a syntax error message, you will see the 67XX Synthesizer respond to the program code.
- After you are satisfied that the program runs correctly, you can use the **Create.EXE** option, on the File menu, to create a stand-alone DOS executable (*.EXE) file. Alternatively, you can use the **Create RTM** option, on the file menu, to create a run-time version (*.RTM) file. That operation is described in the next heading.
- To continue with this tutorial, save this program as *d*\LW\PROGRAMS\67SAMP.BAS (*d*=drive, A:, B:, C:, etc..)

Creating a Compiled Program

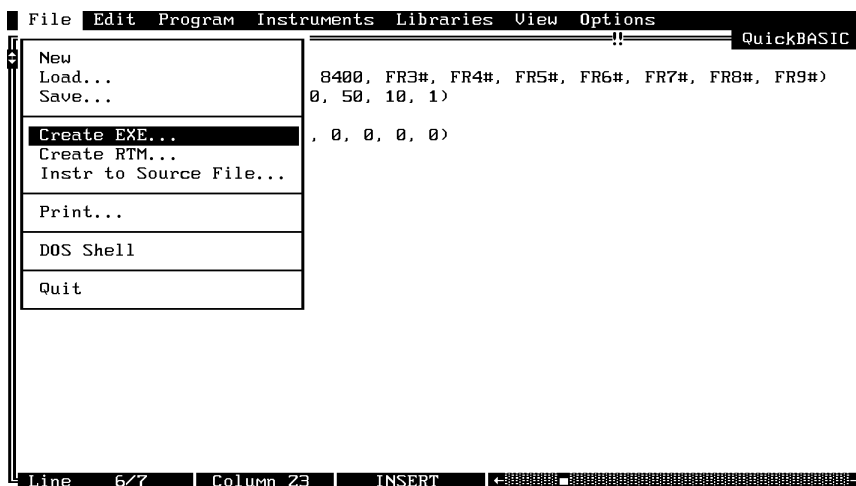
Most programs developed with LabWindows can be compiled with the Microsoft C or BASIC compiler. Some applications, however, become too large to compile and run in the 640 KB DOS memory and must be run within LabWindows or the LabWindows Run-Time System. The LabWindows Run-Time System includes a DOS extender so programs can access up to 16 MB of memory during execution. Programs executed in the run-time system can make calls to any of the LabWindows libraries and instrument drivers. Programs distributed with the run-time system are in a binary format, so the programs cannot be edited.

A stand-alone application (*.EXE or *.RTM) that incorporates the 67XX Driver may be created using the Microsoft C or BASIC compilers or LabWindows Run-Time System. To avoid OUT OF MEMORY errors when using the Microsoft QuickBASIC compiler, you must first optimize the 67XX Driver memory usage with the LabWindows FUNNEL.EXE utility (See *LabWindows User's Manual* for instructions).

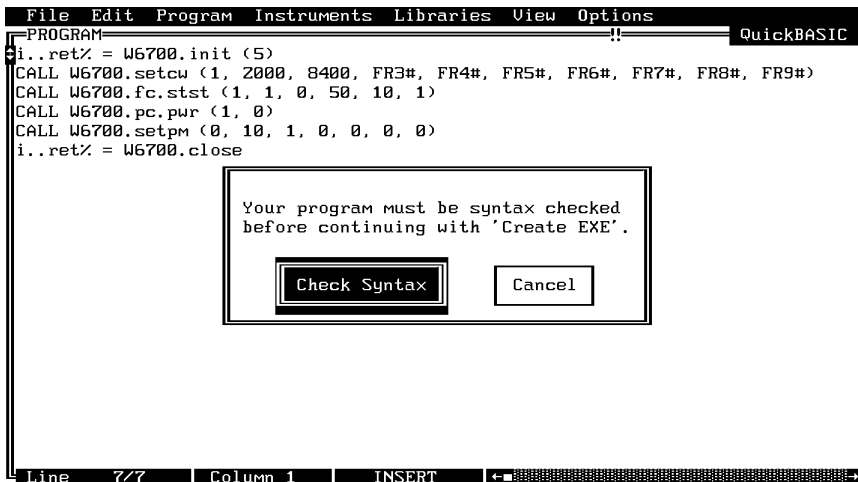
The LWMAKE option on the file menu can be used to create an executable file in either BASIC or C. The following provides a step-by-step tutorial for creating a *.EXE file using the BASIC compiler. This tutorial assumes that you have read and are familiar with the LWMAKE utility description in the *Lab Windows User's Manual*.

We will start with the program that you completed in the preceding tutorial. If you did not complete the tutorial, you can type the program listing shown on the preceding page, and save it as *d\LW\PROGRAMS\67SAMP.BAS*.

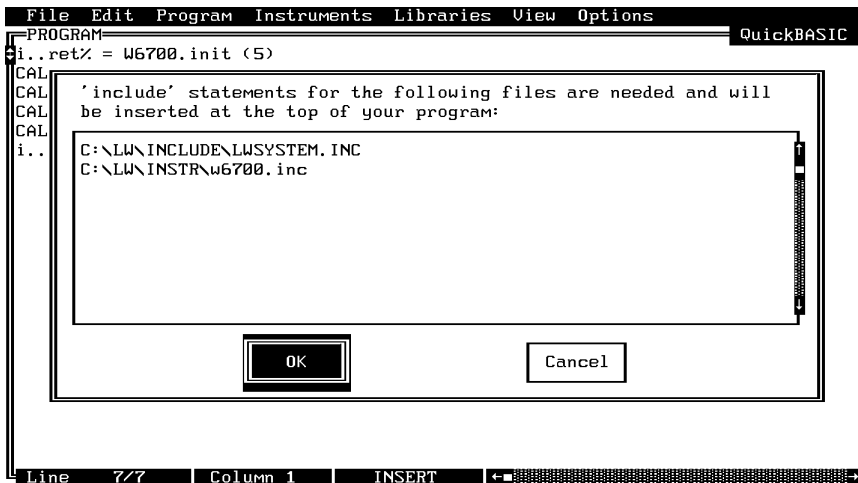
- Move the cursor to **Instruments**, in the top menu bar, and ensure that the **Wiltron 67XX Synthesizer** driver is loaded. If it is not loaded, refer to pages 2-4 and 2-5 for instructions.
- Move the cursor to **File**, in the top menu bar, and select **Create EXE** from the pull-down menu (below).



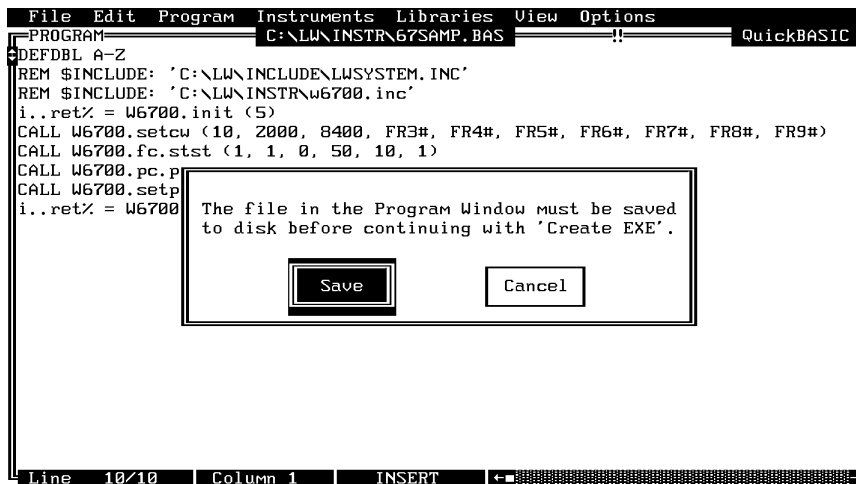
- Choose **Check Syntax**, when the next prompt appears. This will insure that the program does not contain any errors. (*NOTE: This menu may not appear, depending upon previous menu selections.*)



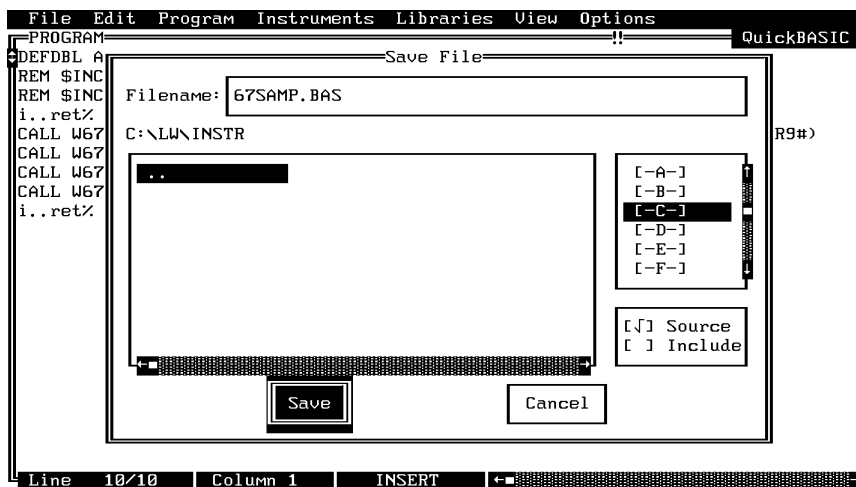
- Answer **OK** to the next prompt. This will include calls to the two required files shown in the file list. These files contain code needed to run your application.



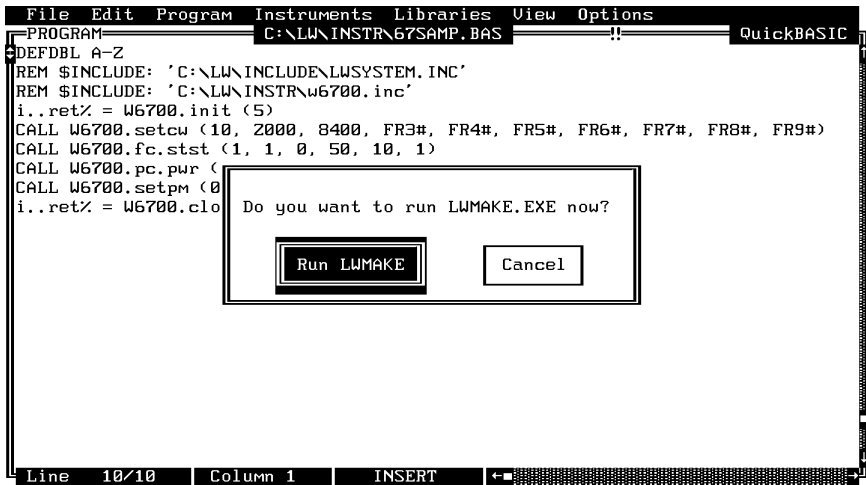
- Choose **Save** for the next prompt, below.



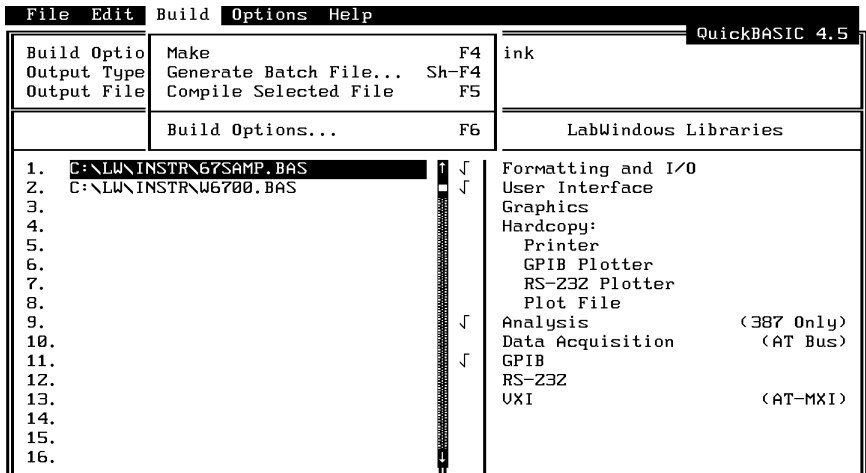
- Choose **Save**, again, to save the changes to the file.



- Choose **Run LWMAKE**, in the next prompt.



- Move cursor to **Build** and choose **Make** from the pull-down menu. (Note: This assumes that the compile, link, and output paths have been previously defined under the LWMAKE Options menu.)



- At the conclusion of the processing that occurs next, the DOS executable file 67SAMP.EXE will appear in the subdirectory containing the like-named BASIC file. Press <ENTER> to return to the LWMAKE screen. To return to the LabWindows PROGRAM window, select **QUIT-Return to LabWindows** from the **File** menu.
- To check your handiwork, choose **DOS SHELL** from the **File** menu. At the ensuing DOS prompt, type 67SAMP to run the program.

Section 3

Driver References

Section 3

Driver References

Introduction

This section lists all the 67XX Instrument Driver function calls in alphabetical order.

Function Panel Structure

The 67XX LabWindows Driver contains twenty panels that provide an intuitive method for coding instrument functions. Each of the panels provide coding for one or more array variables. Table 3-1 list the functional panel in alphabetical order and shows the page number in which they appear. Figure 3-2 does the same for the variables that are used with the driver. Figure 3-1 provides a tree that shows the hierachial structure of the functional panels.

Functional Panel Descriptions

The 67XX driver contains twenty functional panels. These panels appear on the following pages. They are arranged in alphabetical order by call sign (e.g., fc.als; pc.pwr; get-meas; etc.).

Table 3-1. Functional Panels

Panel Name	Page No.	Panel Name	Page No.
Alternate Sweep	3-6	Output Power	3-26
AM Modulation	3-30	Output Power Sweep	3-28
Close	3-46	Pulse Modulation	3-38
Delta Sweep	3-8	Recall Setup	3-40
FM Modulation	3-10	Save Setup	3-42
Frequency Setup	3-34	Source Frequency Def	3-22
Initialize	3-32	Special Sweep	3-14
Marker Selection	3-20	Start/Stop Sweep	3-16
Measurement Functions	3-18	Utilities	3-44
Output Leveling	3-24		

67XX LabWindows Driver User's Guide

Table 3-2. Variables Used With 67XX Driver

Variable	Function Name	Page	Variable	Function Name	Page
ADD%	GPIB Address	3-19	MKSEL%	Marker Select	3-35
ALC%	ALC Output	3-23	MODS%	Modulation Source	3-37
ALTS%	Alternate Sweep	3-7			
AMD#	AM Depth	3-17	NUMBER OF		
			FREQ-		
BLNKP%	Blanking Pulse	3-43	UENCIES%	No. of Freq	3-13
			NUMST%		
CFREQ%	Center Frequency	3-11	NUMST%	Number of	3-7,
COUPL%	Coupling of External			Steps for PL	3-11,
	Source	3-29		Sweeps	3-15
CWFREQ%	Frequency of Operation	3-8	NUMST%	Number of	
CWR%	Rear Panel CW Ramp			Steps, Level	
	Out	3-43		1 to Level 2	3-27
			OPT%	Output Power	3-25
DPM%	Delta Power Meter				
	Mode	3-17	PLR%	Pen Lift Relay	3-43
DPTS%	Number of Frequencies to		PM%	Power Meter	
	be entered or recalled	3-13		Measurement	3-17
DSEL%	Data Select	3-13	PMR#	Power Meter Reading	3-17
DTIME#	Delay Time	3-11, 3-15	PP#	Pulse Period of	
	from External Gate	3-37		Internal PPG	3-37
DTIME#	Phase Locked Dwell	3-7,	PW#	Pulse Width of	
	Time	3-15		Internal PPG	3-37
DWT#	Dwell Time at Each Step	3-27	PWRL#	Output Power Level	3-25
			PWRMTR:		
EDEL%	External Gate Delay	3-37	EXG%	External Gain Cal	3-23
EXDAC%	External DAC Gain	3-23			
			REG%	Storage Register,	3-39,
F1% THRU				1 thru 9	3-41
F9%	CW Frequency Active	3-21	RFDR%	RF During Retrace	3-43
F1% THRU			RFFS%	RF During Band-	
F9%	Markers, 1 thru 9	3-35		switching	3-43
FMD#	FM Depth	3-17	RTN#	Frequency Returned	
FON%	AM On	3-29, 3-33		from 67XX	3-8
FPD%	Front Panel Display	3-43			
FREQSEL%	Selected Frequencies	3-31	SENS#	AM Sensitivity	3-29
FSPAN#	Frequency Span (MHz)	3-11	SENS#	FM Sensitivity	3-33
FX#	Frequency to be Loaded		SWEEP%	Sweep Type	3-7, 3-11, 3-15
	in Storage Register	3-31	SWEEPT#	Analog Sweep	
				Time	3-7, 3-11, 3-15
HRE%	High Resolution Mode	3-43	SWRANGE%	Sweep Range	3-15
LEV%	Leveling	3-23	TEST\$	Filename	3-13
LEVOFF%	Level Offset	3-23	TRIGG%	Trigger	
LKM%	Lock Mode	3-33		Conditions	3-7, 3-11, 3-15
LVC%	Level Correction	3-23			
LVL1#	Level 1 Output Power	3-27	XDGP%	External Gate	
LVL2#	Level 2 Output Power	3-27		Polarity	3-37
LVO%	Level Offset On/Off	3-23	XPG%	External PPG Gate	3-37
MAM%	Measure AM	3-17			
MFM%	Measure FM	3-17			

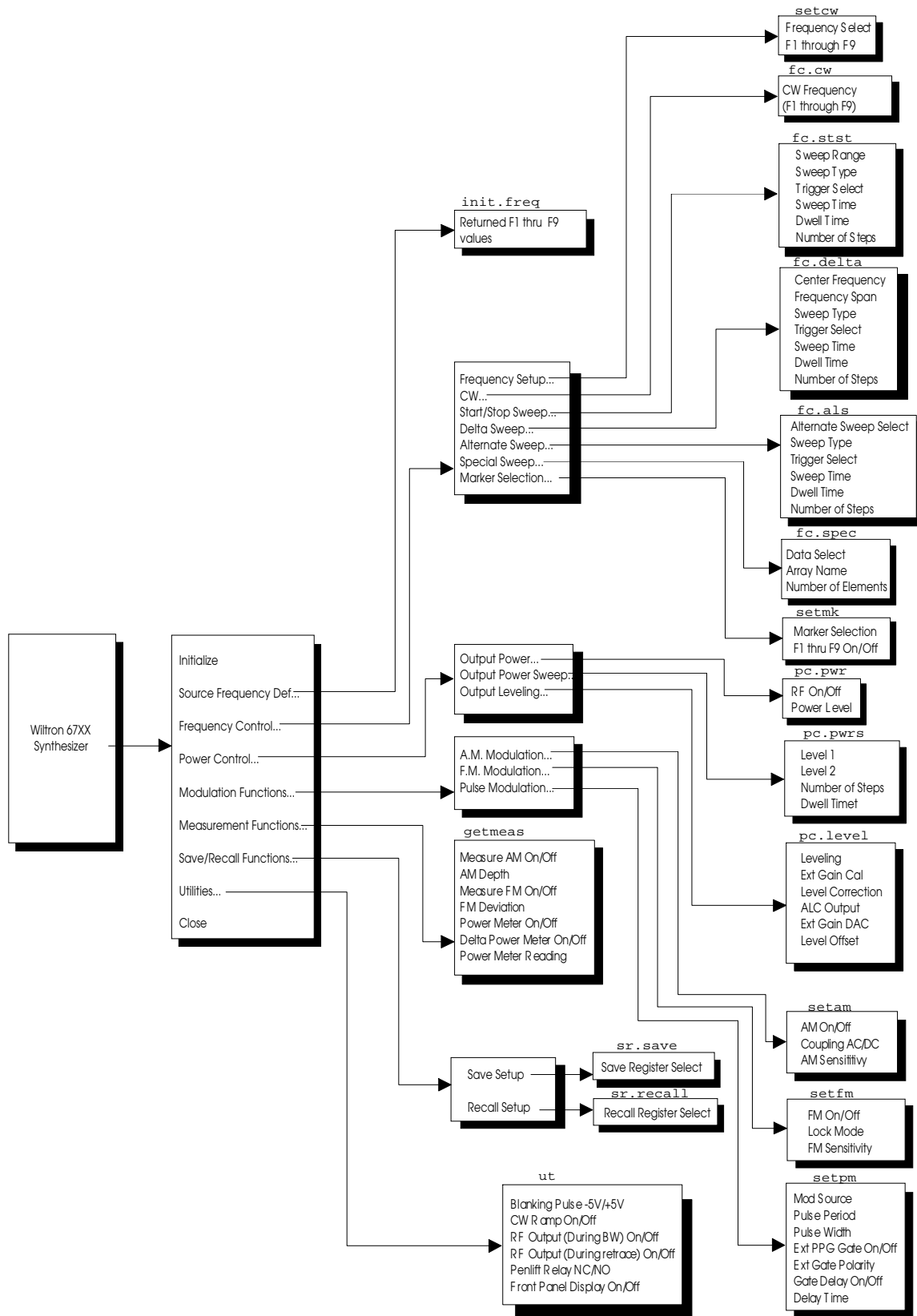
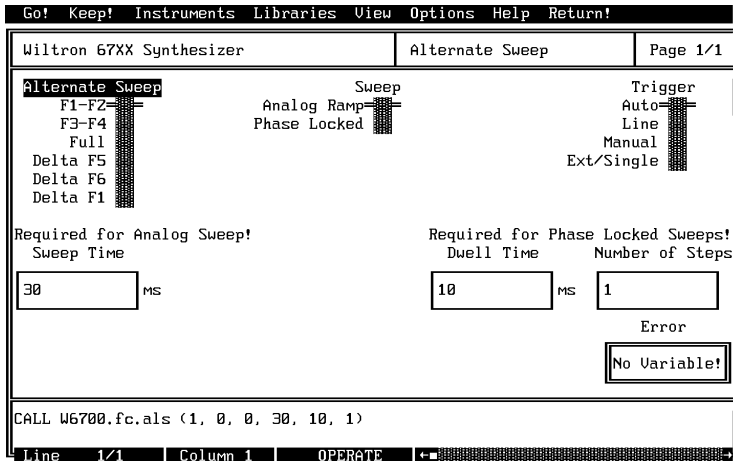


Figure 3-1. Function Panel Tree

fc.als

Function Panel Name: Alternate Sweep

Description: This function lets users alternate consecutive sweeps with pre-existing frequency sweeps. The source must be sweeping prior to using this function or no action is taken.



Controls:

- Alternate Sweep:** Lets users select the sweep range that will alternate with the primary sweep: F1–F2, F3–F4, Full, Delta F5 (Δ F5), Delta F6 (Δ F6), Delta F1 (Δ F1).
- Sweep:** Lets users select an analog or a phase-locked (digitally stepped) alternate sweep.
- Trigger:** Lets users select the triggering source for the alternate sweep:
 - Auto:** Sweep is triggered automatically by the 67XX.
 - Line:** Sweep is triggered from the 67XX ac line.
 - Manual:** Sweep can be triggered manually from the 67XX front panel.
 - Ext/Single:** Sweep can be triggered manually by pressing the SINGLE SWEEP button on 67XX front panel or by supplying a TTL pulse to the rear panel SWEEP TRIGGER INPUT connector.
- Sweep Time:** Lets users enter the time required for one analog sweep (no entry required for phase-locked sweep).

Dwell Time: Lets users enter the dwell time between adjacent frequency steps for a phase-locked sweep (no entry required for analog sweep).

Number of Steps: Lets users enter the number of frequency steps for a phase-locked sweep (no entry required for analog sweep).

Input Parameters: (D) is default setting.

Variable Name	Variable Type	Description	Details	Range
ALTS%	Integer	Alternate Sweep	1, F1–F2 (D) 2, Full Band 3, F3–F4 4, Delta F1 5, Delta F5 6, Delta F6	N/A
SWEEP%	Integer	Sweep Type	0, Analog Ramp (D) 1, Phase Locked	N/A
TRIGG%	Integer	Trigger Conditions	0, Auto (D) 1, Line 2, Ext/Single 3, Manual	N/A
SWEEPT#	Real	Analog Sweep Time	30 ms (D)	30 ms to 99 s.
DTIME#	Real	phase-locked Dwell Time	10 ms (D)	1 ms to 99 s.
NUMST%	Integer	Number of Steps For Phase Locked Sweeps	1 (D)	1 to 1800 steps.

Output Parameters: None.

Error: LabWindows error codes only (220-300)

Program Examples:

Quick BASIC:

```
REM Selects sweep to alternate with main sweep.
CALL V6700.f.c.al s (ALTS% SWEEP% TRI GG% SWEEPT#, DTI ME#, NUMST%)
```

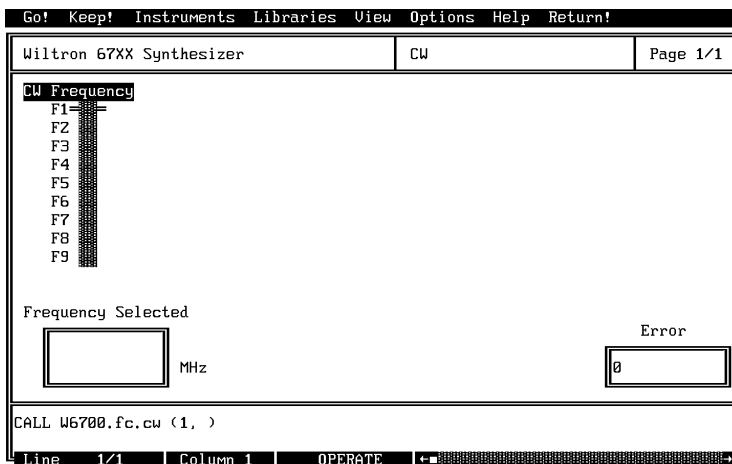
Microsoft C:

```
/* Selects sweep to alternate with main sweep. */
V6700_f.c.al s (ALTS, SWEEP, TRI GG, SWEEPT, DTI ME, NUMST)
```

f c. CW

Function Panel Name: CW

Description: This function lets users select the desired CW output frequency from the values already stored in the 67XX frequency registers (F1 thru F9).



Controls:

CW Frequency: Lets users select CW output frequency from one of the user-defined frequency registers: F1 through F9 (see set cw function, page 3-32).

Frequency Selected: Returns the CW frequency currently stored in the selected register, in MHz.

Input Parameters:

Variable Name	Variable Type	Description	Details	Range
CWFREQ%	Integer	The frequency of operation	1-9	N/A
RTN#	Real	Frequency returned from 67XX.		N/A

Output Parameters: None.

Error: LabWindows error codes only (220-300)

Program Examples:

Quick BASIC:

```
REM Select a CWfrequency and value.  
CALL V6700.CW ( CWFREQ%, RTN#)
```

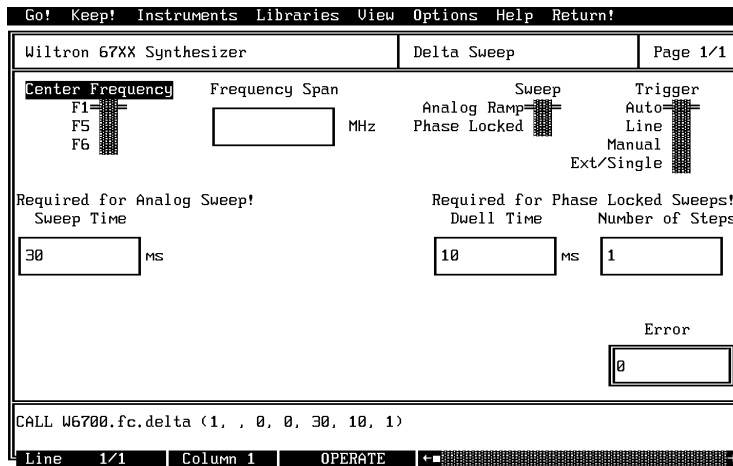
Microsoft C:

```
/* Select a CWfrequency and value. */  
V6700_CW ( CWFREQ, RTN)
```

fc. del t a

Function Panel Name: Delta Sweep

Description: This function lets users set a center frequency and span for the sweep range. The center frequency must be located in storage register 1, 5, or 6.



Controls:

- Center Frequency:** Lets users select (Delta- F (ΔF_n) center frequency) a frequency storage location: 1, 5, or 6.
- Frequency Span:** Lets users select a sweep width (span) in MHz. The 67XX will sweep upwards in frequency from $Center - (span / 2)$ to $Center + (span / 2)$.
- Sweep:** Lets users select either an analog or a phase-locked digitally stepped sweep.
- Trigger:** Lets users select the triggering source for the delta sweep:
 - Auto:** Sweep is triggered automatically by the 67XX.
 - Line:** Sweep is triggered from the 67XX ac line.
 - Manual:** Sweep triggered manually from 67XX front panel.
 - Ext/Single:** Sweep triggered manually using SINGLE SWEEP button on 67XX front panel or by supplying a TTL pulse to rear SWEEP TRIGGER INPUT connector.
- Sweep Time:** Lets users enter the time required for one analog sweep.
- Dwell Time:** Lets users enter the dwell time for a phase-locked sweep.

Number of Steps: Lets users enter the number of frequency steps for a phase-locked sweep.

Input Parameters: (D) is default setting.

Variable Name	Variable Type	Description	Details	Range
CFREQ%	Integer	Center Frequency	1, F1 (D) 5, F5 6, F6	
FSPAN#	Real	Frequency Span(MHz)		(Analog Sweep) 1 MHz to full range of 67XX. (Digital Sweep) 1 kHz to full range of 67XX.
SWEEP%	Integer	Sweep Type	0, Analog Ramp (D) 1, Phase Locked	
TRIGG%	Integer	Trigger Conditions	0, Auto (D) 1, Line 2, Ext/Single 3, Manual	
SWEPT#	Real	Analog Sweep Time	30 mSec (D)	30 ms to 99 s.
DTIME#	Real	PhaseLocked Dwell Time	10 mSec (D)	10 to 99 s.
NUMST%	Integer	Number of Steps For Phase Locked Sweeps	1 (D)	1 to 1800 steps

Output Parameters: None.

Error: LabWindows error codes only (220-300)

Program Examples:

Quick BASIC:

```
REM Selects delta sweep.
CALL V6700.f.c.delta (CREQ% FSPAN#, SWEEP% TRIGG% SWEPT#, DTIME#,
NUMST%)
```

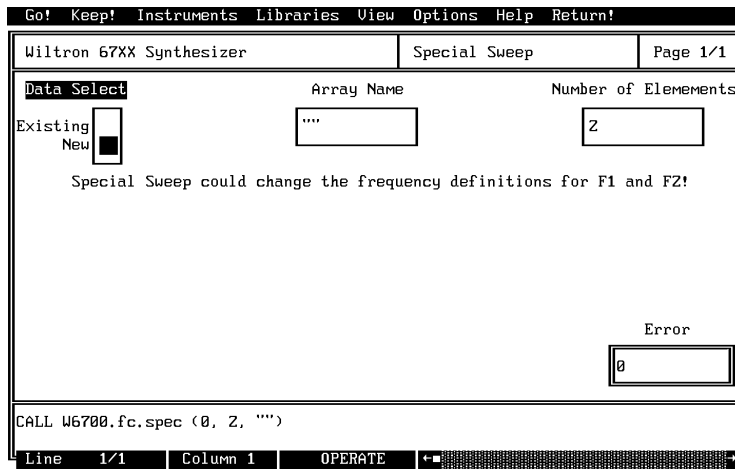
Microsoft C:

```
/* Selects delta sweep. */
V6700_f_c_delta (CREQ, FSPAN, SWEEP, TRIGG, SWEPT, DTIME, NUMST);
```

f c. spec

Function Panel Name: Special Sweep

Description: This function lets users specify frequencies and the order that the frequency will be output. The operator can specify up to 501 discrete frequencies.



Controls:

- Data Select:** Lets users select between new and existing *.wsw (disk file) data.
- Array Name:** Lets users enter the name of the data-array holding the new or existing data. If no array name is entered, the special sweep will still be generated. However, it will not be stored to disk in the current default directory.
- Number of Elements:** Enter the number of frequency elements (steps) for a user-defined special sweep.

Input Parameters: Same as controls.

Variable Name	Variable Type	Description	Details
DSEL%	Integer	Data select	0 = New data file 1 = Existing data file
DPTS%	Integer	Number of frequencies to be entered or recalled.	
TEST\$	String	Filename	If no file name is entered the special sweep is still generated but not saved. The driver automatically appends a ".wsw" extension to the user-entered filename. The file is saved to the current default directory.

Output Parameters: None.

Error: LabWindows error codes only (220-300). If debug W6700.err% errors reported (3XX).

Program Examples:

Quick BASIC:

```
REM Lets operator enter frequencies in GHzertz for a user-defined
(special) sweep.
CALL W6700.f.c.spec (DSEL% DPTS% TEST$)
```

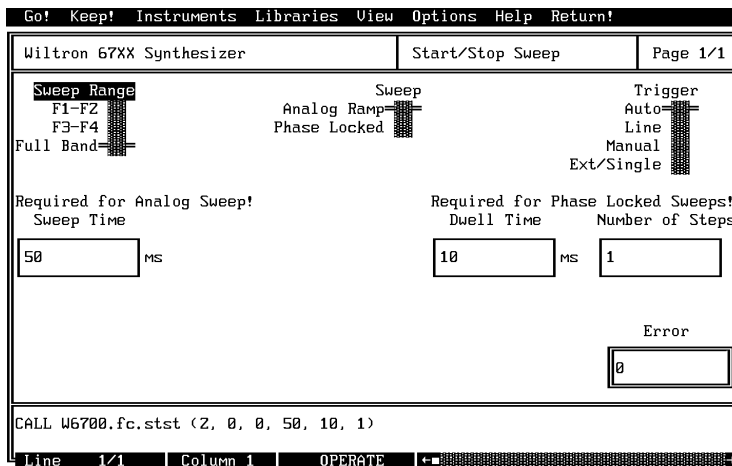
Microsoft C:

```
/* Lets operator enter frequencies in GHzertz for a user-defined
(special) sweep. */
W6700_f_c_spec (DSEL, DPTS, TEST)
```


f c. st st

Function Panel Name: Start/Stop Sweep

Description: This function lets users select the 67XX sweep range. The function uses the existing definitions for CW frequencies in the F1 through F9 registers. The user has the opportunity to select the source trigger conditions and set sweep time — if in analog sweep. Or they can select dwell time and number of steps — if in phase locked sweep.



Controls:

- Sweep Range:** Lets users select the primary sweep range: F1–F2, F3–F4, or Full band.
- Sweep:** Lets users select an analog or a phase-locked (digitally stepped) primary sweep.
- Trigger:** Lets users select the triggering source for the primary sweep:
 - Auto:** Sweep is triggered automatically by the 67XX.
 - Line:** Sweep is triggered from the 67XX ac line.
 - Manual:** Sweep can be triggered manually from the 67XX front panel.
 - Ext/Single:** Sweep can be triggered manually by pressing the SINGLE SWEEP button on 67XX front panel or by supplying a TTL pulse to the rear panel SWEEP TRIGGER INPUT connector.
- Sweep Time:** Lets users enter the time required for one analog sweep (no entry required for phase-locked sweep).

Dwell Time: Lets users enter the dwell time between adjacent frequency steps for a phase-locked sweep (no entry required for analog sweep).

Number of Steps: Lets users enter the number of frequency steps for a phase-locked sweep (no entry required for analog sweep).

Input Parameters: (D) is default setting.

Variable Name	Variable Type	Description	Details	Range
SWRANGE%	Integer	Sweep Range	1, F1–F2 2, Full Range (D) 3, F3–F4	N/A
SWEEP%	Integer	Sweep Type	0, Analog Ramp (D) 1, Phase Locked	N/A
TRIGG%	Integer	Trigger Conditions	0, Auto (D) 1, Line 2, Ext/Single 3, Manual	N/A
SWEPT#	Real	Analog Sweep Time	30 mSec (D)	30 ms to 99 s
DTIME#	Real	PhaseLocked Dwell Time	0 mSec (D)	1 ms to 99 s.
NUMST%	Integer	Number of Steps For Phase Locked Sweeps	1 (D)	1 to 1800 steps.

Output Parameters: None.

Error: LabWindows error codes only (220-300)

Program Examples:

Quick BASIC:

```
REM Selects a primary (main) sweep.
CALL V6700.f.c.stst (SWRANGE% SWEEP% TRIGG% SWEPT#, DTIME#, NUMST%)
```

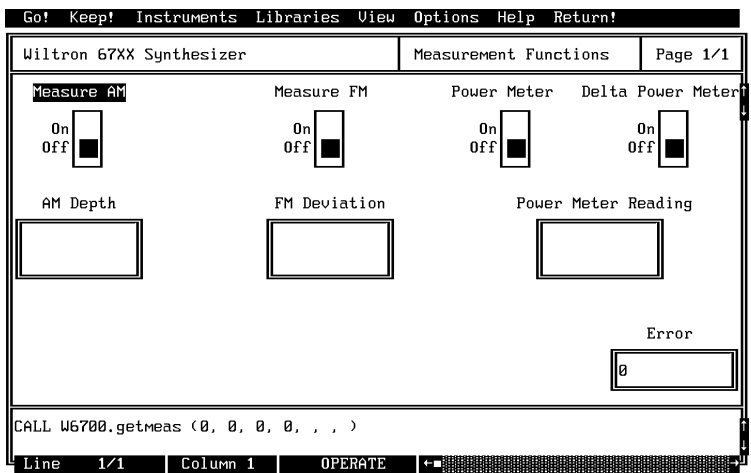
Microsoft C:

```
/* Selects a primary (main) sweep. */
V6700_f.c.stst (SWRANGE, SWEEP, TRIGG, SWEPT, DTIME, NUMST)
```

get meas

Function Panel Name: Measurement Function

Description: This function gives users access to the measurement functions available in the 67XX: AM/FM characteristics and power.



Controls:

Measure AM: Lets users turn the Measure AM function on or off. This function measures the voltage of the external modulation signal and calculates the percentage modulation value. The result is displayed in the AM Depth field.

Measure FM: Lets users turn the Measure FM function on or off. This function measures the voltages of the external modulation signal and calculates the peak frequency deviation. The result is displayed in the FM Deviation field.

Power Meter: Lets users turn the Power Meter function on and off. This function allows the power to be measured with a WILTRON 560-7 Series Detector connected to the rear panel POWER METER connector.

Delta Power Meter: Lets users turn the Delta Power Meter function on or off. Using this function the power measured at the time the function is activated establishes a 0 dBm reference level. Subsequent changes in measured power are indicated with respect to the reference.

AM Depth: Displays the measured AM depth value.

FM Deviation: Displays the measured FM deviation value.

Power Meter Reading: Displays the measured power, either absolute or delta, depending on control selections.

Input Parameters: (D) is default setting.

Variable Name	Variable Type	Description	Details	Range
MAM%	Integer	Measure AM	0, Off (D) 1, On	N/A
MFM%	Integer	Measure FM	0, Off (D) 1, On	N/A
PM%	Integer	Power Meter Measurement	0, Off (D) 1, On	N/A
DPM%	Integer	Delta Power Meter Mode	0, Off (D) 1, On	N/A
AMD#	Real	AM Depth	Returned value	N/A
FMD#	Real	FM Depth	Returned value	N/A
PMR#	Real	Power Meter Reading	Returned value	N/A

Output Parameters: None.

Error: LabWindows error codes only (220-300)

Program Examples:

Quick BASIC:

```
REM Provides for using modulation measurement functions in source.
CALL V6700.get meas (MAN%, MFM%, PM%, DPM%, AMD#, FMD#, PMR#)
```

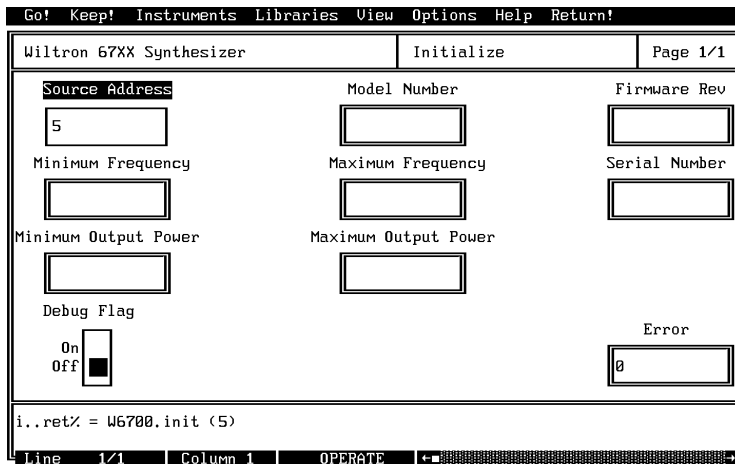
Microsoft C:

```
/* Provides for using modulation measurement functions in source. */
V6700_get_meas (MAN, MFM, PM, DPM, AMD, FMD, PMR)
```

init

Function Panel Name: Initialize

Description: This function opens the GPIB communication with the 67XX and sets it to its preset state. The device configuration must already be completed.



Controls:

- Source Address:** Lets users enter a GPIB address for the 67XX. The factory-set default address is 5.
- Model Number:** Displays the 67XX model number.
- Firmware Rev:** Displays the firmware revision level.
- Minimum Frequency:** Displays the minimum frequency for a full band sweep.
- Maximum Frequency:** Displays the maximum frequency for a full band sweep.
- Serial Number:** Displays the 67XX serial number.
- Minimum Output Power:** Displays the minimum output power to which the 67XX can be set.
- Maximum Output Power:** Displays the maximum output power to which the 67XX can be set.
- Debug Flag:** Lets users turn the Debut Utility on or off. This utility is discussed on page 2-6.

Input Parameters: Same as controls.

Variable Name	Variable Type	Description	Details	Range
ADD%	Integer	GPIB Address	1-9, 5 (D)	1 to 31

Output Parameters: None.

Error: LabWindows error codes only (220-300)

Program Examples:

Quick BASIC:

```
REM Initializes the 67XX.
CALL V6700_init (ADD%)
```

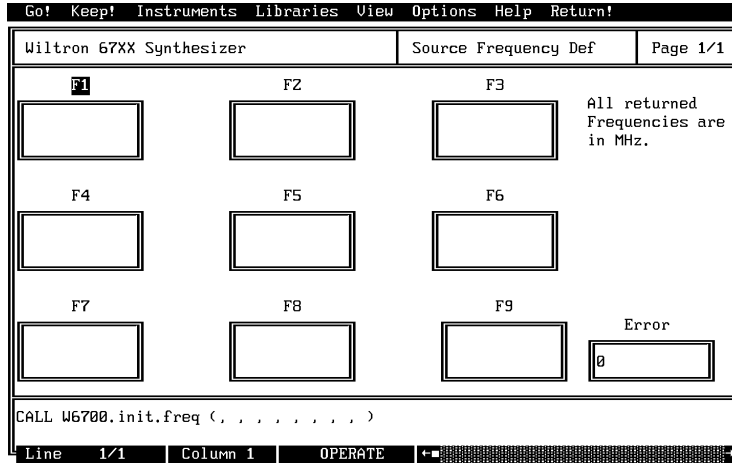
Microsoft C:

```
/* Initializes the 67XX. */
V6700_init (ADD)
```

init.freq

Function Panel Name: Source Frequency Def(inition)

Description: This function returns the values of the frequencies (F1-F9) that are presently stored in the 67XX storage registers. All frequencies are returned in megahertz. There are no operator inputs.



Controls:

- F1 :** Displays the 67XX F1 frequency.
- F2 :** Displays the 67XX F2 frequency.
- F3 :** Displays the 67XX F3 frequency.
- F4 :** Displays the 67XX F4 frequency.
- F5 :** Displays the 67XX F5 frequency.
- F6 :** Displays the 67XX F6 frequency.
- F7 :** Displays the 67XX F7 frequency.
- F8 :** Displays the 67XX F8 frequency.
- F9:** Displays the 67XX F9 frequency.

Input Parameters: Same as controls.

Variable Name	Variable Type	Description	Details	Range
F1# thru F9#	Real	CW frequencies active in the storage registers.	N/A	

Output Parameters: None.

Error: LabWindows error codes only (220-300)

Program Examples:

Quick BASIC:

```
REM Get the value of up to six real and imaginary markers.
CALL W6700_init_freq ( F1#, F2#, F3#, F4#, F5#, F6#, F7#, F8#, F9#)
```

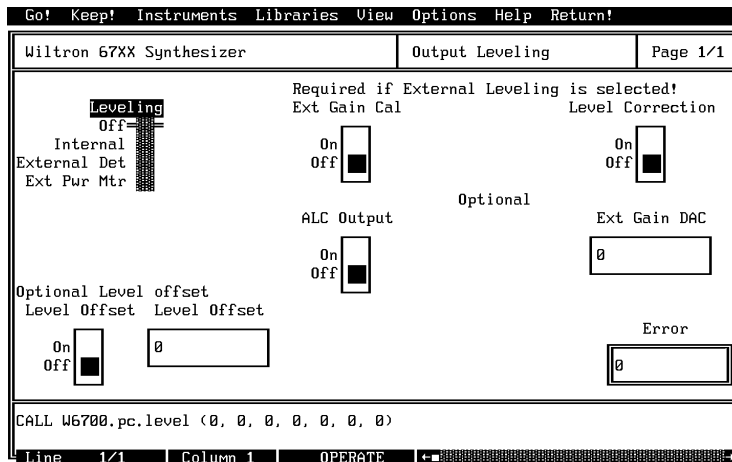
Microsoft C:

```
/* Get the value of up to six real and imaginary markers. */
W6700_init_freq ( F1, F2, F3, F4, F5, F6, F7, F8, F9)
```


pc. l level

Function Panel Name: Output Leveling

Description: This function lets users select the power leveling mode(s).



Controls:

- Leveling** Lets users select the leveling mode for the output power.
- Off:** No leveling is applied.
- Internal:** Output power is leveled using an internal directional coupler to sample the output signal.
- External Det:** Output power is leveled using an external directional coupler to sample the output signal.
- Power Meter:** Output power is leveled by using the “recorder voltage” from an external power meter. The 67XX is compatible with power meters having a ± 1 V full scale analog output.
- Ext Gain Cal:** Automatically adjusts the gain for the leveling signal from an external detector to stabilize the ALC loop.
- Level Correction:** Lets users turn the internal level vs. frequency correction on or off.
- ALC Output:** Outputs the voltage from the ALC linearizer if turned on.
- Ext Gain DAC:** Allows adjustment of the external-level-gain DAC to optimize loop stability.

Level Offset:

(Switch): Turns the optional level-offset function on or off.

(Value Field): Lets users enter a value (dB) to offset the displayed output power level.

Input Parameters: (D) is default setting

Variable Name	Variable Type	Description	Details	Range
LEV%	Integer	Leveling	0, Off (D), 1, Internal 2, Ext Det 3, Ext	N/A
PWRMTR: EXG%	Integer	Ext Gain Cal	0, Off (D) 1, On	N/A
LVC%	Integer	Level Correction	0, Off (D) 1, On	N/A
ALC%	Integer	ALC Output	0, Off (D) 1, On	N/A
EXDAC%	Real	Ext Dac Gain	0 (D)	0 to 255
LEVOFF%	Real	Level Offset	Value of Power Offset	???
LVO%	Integer	Level Offset On/Off	0, Off (D) 1, On	N/A

Output Parameters: None.

Error: LabWindows error codes only (220-300)

Program Examples:**Quick BASIC:**

```
REM Select the leveling mode.
CALL V6700.pc_level (LEV%, EXG%, LVC%, ALC%, EXDAC#, LEVOFF#, LVO%)
```

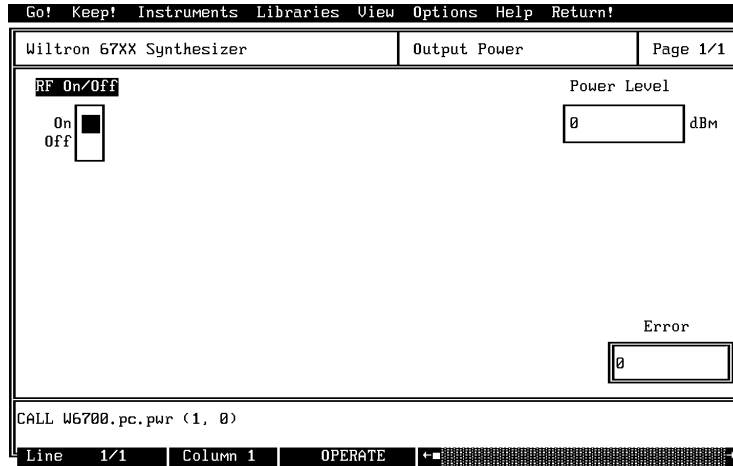
Microsoft C:

```
/* Select the leveling mode. */
V6700.pc_level (LEV, EXG, LVC, ALC, EXDAC, LEVOFF, LVO)
```

pc. pw

Function Panel Name: Output Power

Description: This function lets users set the 67XXs output power level.



Controls:

RF On/Off: Lets users turn the RF output power on and off.

Power Level: Lets users enter the desired output power level.

Input Parameters: (D) is default setting.

Variable Name	Variable Type	Description	Details	Range
OPT%	Integer	Output Power	0 = On(D) 1 = Off	N/A
PWRL#	Real	Output Power Level	0 dBm (D)	Depends on particular 67XX model and whether Option 2X Step Attenuator is fitted. Refer to 67XXA or B Operation Manual for details.

Output Parameters: None.

Error: LabWindows errors (220-300). If debug error 301 Parameter out of range.

Program Examples:

Quick BASIC:

```
REM Set out put power .
CALL V6700.pc.pwr (OPT% PWRL#)
```

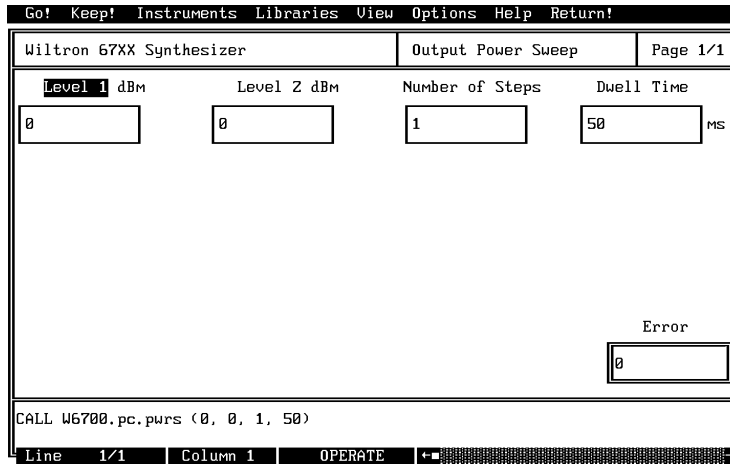
Microsoft C:

```
/* Set out put power. */
V6700_pc_pwr (OPT, PWRL)
```

pc. pwr s

Function Panel Name: Output Power Sweep

Description: This function lets users set up the 67XX in a power sweep mode of operation.



Controls:

Level 1 dBm: Lets users enter the setting in dBm for the start power level value. This is the start value for the power sweep.

Level 2 dBm: Lets users enter the setting in dBm for the stop power level value. This is the stop value for the power sweep.

NOTE

Power sweep can be from high to low or low to high power levels. Therefore, the Level 1 (start) power could be higher or lower than the Level 2 (stop) power.

Number of Steps: Lets users enter the number of discrete power points (steps) at which the sweep will pause (as set using Dwell Time control).

Dwell Time: Lets users enter the length of time they want the power sweep to remain (dwell) at each step.

Input Parameters: (D) is default setting.

Variable Name	Variable Type	Description	Details	Range
LVL1#	Real	Level 1 Output Power	0 dBm (D)	Depends on particular 67XX model and whether Option 2X Step Attenuator is fitted. Refer to 67XXA or B Operation Manual for details.
LVL2#	Real	Level 2 Output Power	0 dBm (D)	Same as above.
NUMST%	Integer	Number of Steps from Level 1 to Level 2	1 step (D)	1 to 1000.
DWT#	Real	Dwell Time At Each Step	50 ms (D)	50 ms to 99 s.

Output Parameters: None.

Error: LabWindows errors (220-300). If debug error 301 Parameter out of range.

Program Examples:

Quick BASIC:

```
REM Define the power sweep.
CALL V6700.pc.pwr s (LVL1#, LVL2#, NUMST% DWT#)
```

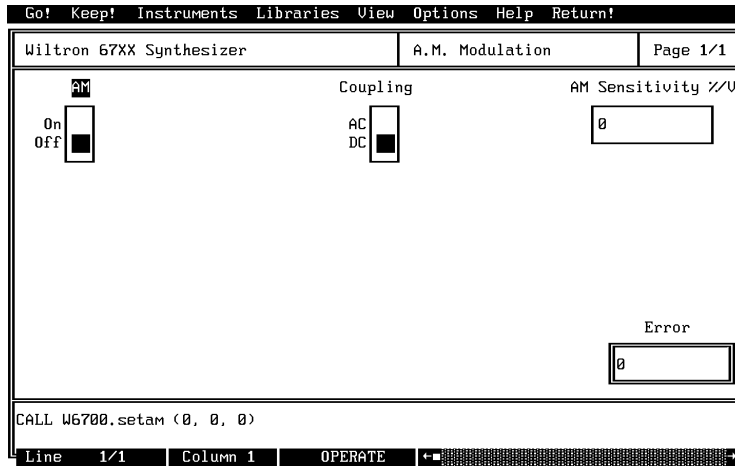
Microsoft C:

```
/* Define the power sweep. */
V6700_pc_pwr s (LVL1, LVL2, NUMST, DWT)
```

set am

Function Panel Name: AM Modulation

Description: This function lets users set amplitude modulation (AM) parameters.



Controls:

AM: Lets users turn AM modulation on and off.

Coupling: Lets users chose ac or dc coupling.

AM Sensitivity %/V: Lets users enter a per-volt sensitivity for the AM modulation that is to be applied to the output signal. Allowable values are from 1 to 99.9 percent.

Input Parameters: (D) is default setting.

Variable Name	Variable Type	Description	Details	Range
FON%	Integer	AM On:	0, Off (D) 1, On	N/A
COUPL%	Integer	Coupling of External Source	0, dc Coupled (D) 1, ac Coupled	N/A
SENS#	Real	AM Sensitivity in Percent/Volt		N/A

Output Parameters: None.

Error: LabWindows errors (220-300).

Program Examples:

Quick BASIC:

```
REM Set modulation values for AM
CALL W6700.set am ( FON% COUP% SENS#)
```

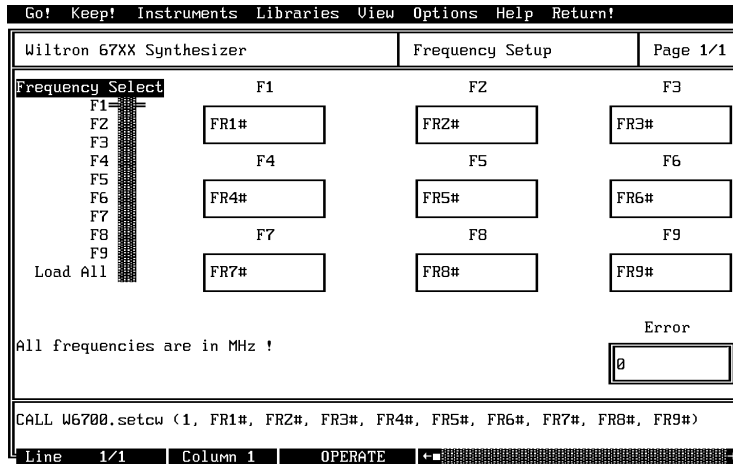
Microsoft C:

```
/* Set modulation values for AM */
W6700_set am ( FON, COUP, SENS)
```


set cw

Function Panel Name: Frequency Setup

Description: This function lets users define from 1 to 9 frequencies for any of the nine 67XX storage registers.



Controls:

Frequency Select: Lets users select any single frequency storage location, or to select all nine at one time. If more than one frequency is to be defined, the switch must be set to the Load All position.

F1 thru F9: Let users enter a frequency value in MHz for storage in the applicable location (register).

Input Parameters:

Variable Name	Variable Type	Description	Range
FREQSEL%	Integer	Selects Frequencies 1–10, where 10 loads all frequencies. If any other value, it selectively loads that value into the corresponding register. If a selective load is used, that frequency then becomes the active output frequency. If load all is selected (10), then frequency register 9 (F9) becomes the active output when the function is complete.	N/A
FX#	Real	Frequency to be loaded into the storage registers.	N/A

Output Parameters: None.

Error: LabWindows error codes only (220-300)

Program Examples:**Quick BASIC:**

```
REM Select a cw frequency storage location and load frequency value.
CALL W6700.set cw ( FREQSEL% F1#, F2#, F3#, F4#, F5#, F6#, F7#, F8#, F9#)
```

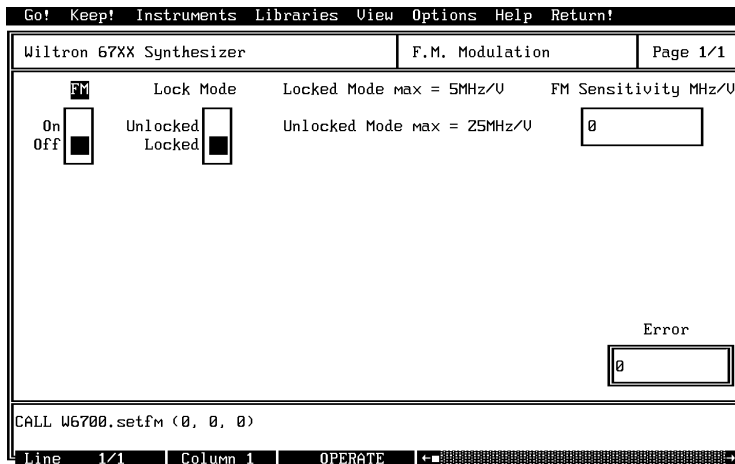
Microsoft C:

```
/* Select a cw frequency storage location and load frequency value. */
W6700_set cw ( FREQSEL, F1, F2, F3, F4, F5, F6, F7, F8, F9)
```

set f m

Function Panel Name: FM Modulation

Description: This function lets users set modulation parameter for frequency modulation using an external modulation source.



Controls:

- FM:** Lets users turn frequency modulation (FM) on and off.
- Lock Mode:** Lets users select between phase locked and unlocked FM modes.
- Unlocked:** Enables the output signal to be modulated by an external FM input applied to either the front panel EXT FM jack or the rear panel FM INPUT connector. In this mode, the 67XX output is phase locked. Deviations are limited to 20 times the rate (20 x rate).
- Locked:** Enables the output signal to be modulated by an external FM input applied to either the front panel EXT FM jack or the rear panel FM INPUT connector. In this mode, the 67XX output is not phase locked. This mode can be used for frequency deviations of up to ±25 MHz.
- FM Sensitivity MHz/V:** Lets users enter the FM sensitivity value.

Input Parameters: (D) is default setting.

Variable Name	Variable Type	Description	Details	Range
FON%	Integer	FM	0, Off (D) 1, On	N/A
LKM%	Integer	Lock Mode	0, Locked (D) 1, Unlocked	N/A
SENS#	Real	FM Sensitivity in MHz/Volt		10 kHz/V to 5 MHz in the locked mode and up to 25 MHz/V in the unlocked mode.

Output Parameters: None.

Error: LabWindows error codes only (220-300).

Program Examples:

Quick BASIC:

```
REM Set up 67XX for external fm
CALL W6700.set fm ( FON% LKM% SENS#)
```

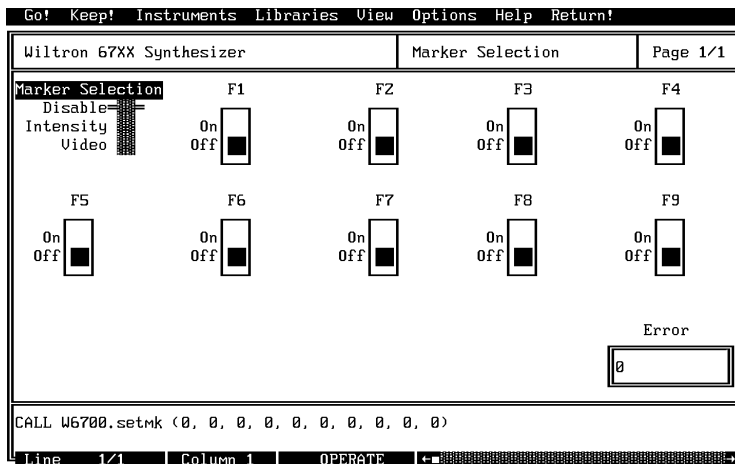
Microsoft C:

```
/* Set up 67XX for external fm */
W6700_set fm ( FON, LKM, SENS)
```

set mk

Function Panel Name: Marker Selection

Description: This function lets users set video and intensity markers.



Controls:

- Marker Selection:** Lets users disable markers or select between Intensity and Video markers.
- Disable:** Disables the frequency markers.
- Intensity:** Enables an intensified dot to mark the frequency, provided the selected marker frequency is within the selected frequency range. (Only available in analog sweep mode.)
- Video:** Enables a video pulse to mark the frequency, provided the selected marker frequency is within the selected frequency range.
- F1 thru F9:** Turns marker on or off at the frequency stored in locations F1 thru F9.

Input Parameters: (D) is default setting.

Variable Name	Variable Type	Description	Details	Range
MKSEL%	Integer	Marker Select	0, Disable (D) 1, Intensity 2, Video	N/A
F1% Thru F9%	Integer	Marker:1-9	0, Off (D) 1, On	N/A

Output Parameters: None.

Error: LabWindows error codes only (220-300)

Program Examples:

Quick BASIC:

```
REM Set marker s.
CALL W6700.set mkr ( MKSEL% F1% F2% F3% F4% F5% F6% F7% F8% F9%)
```

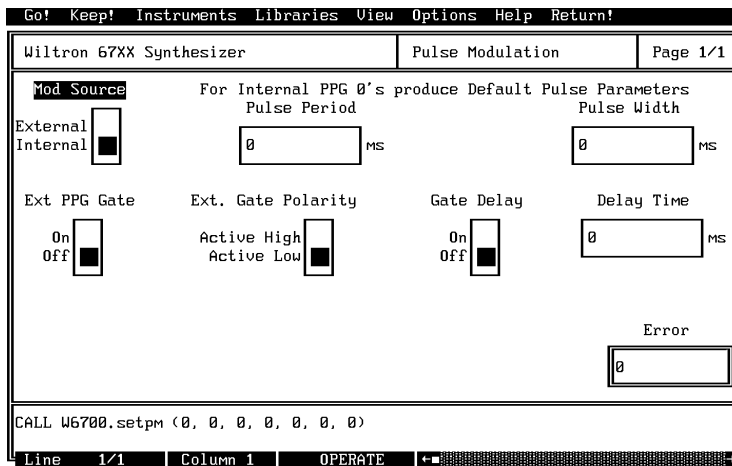
Microsoft C:

```
/* Set marker s. */
W6700_set mkr ( MKSEL, F1, F2, F3, F4, F5, F6, F7, F8, F9)
```

set pm

Function Panel Name: Pulse Modulation

Description: This function lets users set pulse modulation parameters.



Controls:

- Mod Source:** Lets users select a pulse modulation source: External or Internal. The external mode allows an external TTL level trigger the internal pulse generator.
- Pulse Period:** Lets users enter a value in milliseconds for the internal pulse generator period.
- Pulse Width:** Lets users enter a width value in milliseconds for the internal pulse generator pulse.
- Ext PPG Gate:** Lets users turn the external PPG Gate mode on and off. In this mode, an external source can be use to gate the internal pulse generator.
- Ext Gate Polarity:** Lets users select the polarity of the external TTL gating pulse.
- Gate Delay:** Lets users select the delay of the external TTL gating pulse.
- DelayTime:** Lets users enter a delay value in milliseconds.

Input Parameters: (D) is default setting.

Variable Name	Variable Type	Description	Details	Range
MODS%	Integer	Modulation Source	0, Internal (D) 1, External	N/A
PP#	Real	Pulse Period of Internal PPG		0.001 to 100 ms.
PW#	Real	Pulse Width of Internal PPG		0.000025 to 99 ms.
XPG%	Integer	External PPG Gate	0, Off (D) 1, On	N/A
XDGP%	Integer	External Gate Polarity	0, Active Low (D) 1, Active High	N/A
EDEL%	Integer	External Gate Delay	0, Off (D) 1, On	N/A
DTIME#	Real	Delay Time From External Gate		0.000200 to 100 ms.

Output Parameters: None.

Error: LabWindows error codes only (220-300)

Program Examples:

Quick BASIC:

```
REM Set pulse modulation parameters.
CALL V6700.set pm ( MODS% PP#, PW#, XGP% XDGP% EDEL% DTIME#)
```

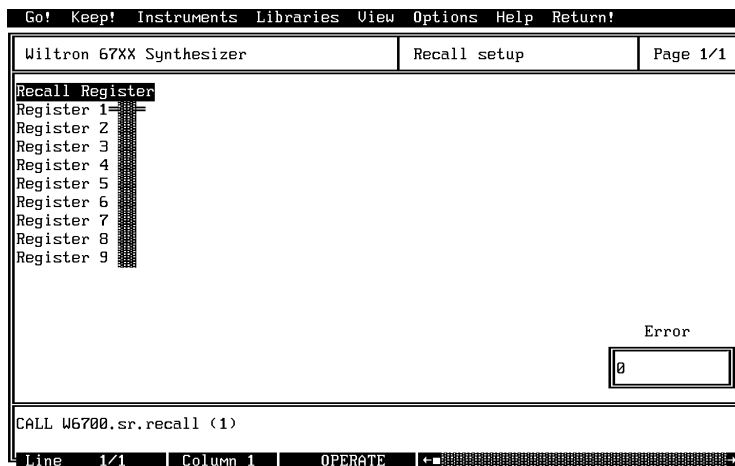
Microsoft C:

```
/* Set pulse modulation parameters. */
V6700_set pm ( MODS, PP, PW XGP, XDGP, EDEL, DTIME)
```


sr . recall I

Function Panel Name: Recall Setup

Description: This function lets users recall an existing source setup from the 67XX internal memory.



Controls:

Recall Register: Lets users select the storage register containing the setup they wish to recall.

Input Parameters: (D) is default setting.

Variable Name	Variable Type	Description	Details	Range
REG%	Integer	Storage Register, 1 Thru 9	1, (D)	N/A

Output Parameters: None.

Error: LabWindows error codes only (220-300)

Program Examples:

Quick BASIC:

```
REM Recal I stored set ups.
CALL W6700.sr.recal I (REG%)
```

Microsoft C:

```
/* Recal I stored set ups. */
W6700_sr.recal I (REG)
```

sr . save

Function Panel Name: Save Setup

Description: This function lets users save an existing setup to the 67XX internal memory.



Controls:

Recall Register: Lets users select a storage register for storing a setup they may wish to recall at a later time.

Input Parameters: (D) is default setting.

Variable Name	Variable Type	Description	Details	Range
REG%	Integer	Save Register, 1 Thru 9	1, (D)	N/A

Output Parameters: None.

Error: LabWindows error codes only (220-300)

Program Examples:

Quick BASIC:

```
REM Recall stored set ups
CALL V6700_sr.save (REG%)
```

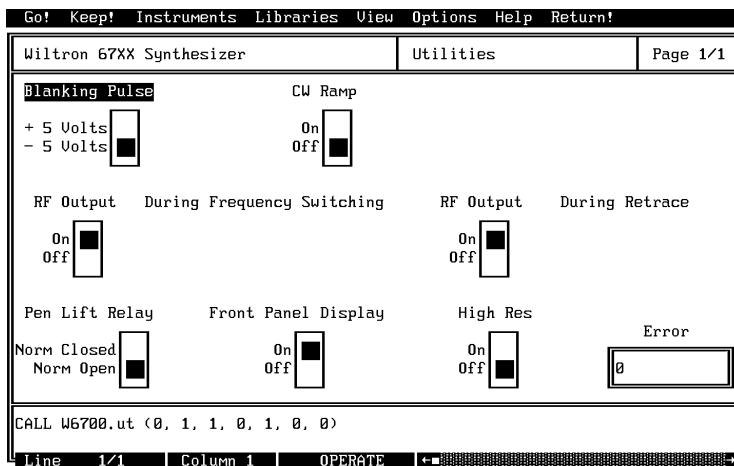
Microsoft C:

```
/* Recall stored set ups */
V6700_sr.save (REG)
```

ut

Function Panel Name: Utilities

Description: This function lets users define many additional functions typically used for advanced applications. It also controls the 67XX during frequency switching and retrace.



Controls:

- Blanking Pulse:** Lets users select rear panel supplied blanking pulse to be ± 5 volts during sweep retrace.
- CW Ramp:** Lets users turn the CW Ramp function on and off. This function causes a 0–10 volt horizontal ramp to be supplied to the rear panel HORIZONTAL OUTPUT connector during CW operations.
- RF Output – During Frequency Switching:** Lets users turn the RF output on and off during band-switching.
- RF Output – During Retrace:** Lets users turn the RF output on and off during sweep re-tracing.
- Pen Lift Relay:** Lets users select between normally open (NO) and normally closed (NC) for the 67XX pen lift relay.
- Front Panel Disp:** Lets users turn the front panel displays on and off.
- High Res:** Lets users turn the High Res Input mode on and off. In this mode, an external 20–32.1 MHz source can be substituted for the internal fine-frequency loop. This will allow for frequency resolution in the millihertz region. External Hi Res input is via a 67XX rear panel connector.

Input Parameters: (D) is default setting.

Variable Name	Variable Type	Description	Details	Range
BLNKP%	Integer	Blanking Pulse	0, -5 Volt (D) 1, +5 Volt	N/A
RFFS%	Integer	RF During Bandswitching	0, Off 1, On (D)	N/A
RFDR%	Integer	RF During Retrace	0, Off 1, On (D)	N/A
PLR%	Integer	Pen Lift Relay	0, Normally Open (D) 1, Normally Closed	N/A
FPD%	Integer	Front Panel Display	0, Off 1, On (D)	N/A
CWR%	Integer	Rear Panel CW Ramp Out	0, Off (D) 1, On	N/A
HRE%	Integer	High Resolution Mode	0, Off (D) 1, On	N/A

Output Parameters: None.

Error: LabWindows error codes only (220-300)

Program Examples:

Quick BASIC:

```
REM Set blanking, CW ramp, RF output (during freq switching and re-
trace), penlift relay, front panel display, and hi resolution values.
CALL V6700.ut (BLNKP% RFFS% RFDR% PLR% FPD% CWR% HRE%)
```

Microsoft C:

```
/* Set blanking, CW ramp, RF output (during freq switching and re-
trace), penlift relay, front panel display, and hi resolution values.
*/
V6700_ut (BLNKP, RFFS, RFDR, PLR, FPD, CWR, HRE)
```

W6700.close

Function Panel Name: Close

Description: This function closes the GPIB communications with the 67XX.



Controls: This panel has no controls.

Input Parameters: None.

Output Parameters: None.

Error: None

Program Examples:

Quick BASIC:

```
REM cl ose  
CALL W6700.W6700.cl ose
```

Microsoft C:

```
/* cl ose */  
W6700_W6700_cl ose
```


Subject Index

Index

!
*.EXE 2-3

A
ADD% 3-19
ALC% 3-23
Alternate Sweep 3-6
ALTS% 3-7
am modulation 3-28
AMD# 3-17

B
BASIC 1-4, 2-17
BLNKP% 3-43

C
C 1-4, 2-17
CFREQ% 3-11
close 3-44
COUPL% 3-29
Creating a Compiled Program 2-17
cw frequency 3-8
CWFREQ% 3-8
CWR% 3-43

D
debug flag 2-10
debug switch 2-8
Debug Utility 2-6
delta sweep 3-10
DPM% 3-17
DTIME# 3-7, 3-11, 3-15, 3-37,
DWT# 3-27

E
EDEL% 3-37
error messages 2-8
EXDAC% 3-23
executable (file) 2-3

F
F1% THRU F9% 3-21
F1% THRU F9% 3-35
fc.als 3-6
fc.cw 3-8
fc.delta 3-10
fc.spec 3-12
fc.stst 3-14
fm modulation 3-32
FMD# 3-17
FON% 3-33
FON% 3-29
FPD% 3-43
FREQSEL% 3-31
frequency setup 3-30
FSPAN# 3-11
Function Panel Structure 3-3
Functional Panel Descriptions 3-3
FX#
Storage Register 3-31

G
General 1-3, 2-3
getmeas 3-16

H
HRE% 3-43

I
init 3-18
init.freq 3-20
initializer 3-18
Installing Instrument Drivers 1-4
Introduction 2-3, 3-3

L
LabWindows libraries 1-4
LEV% 3-23
LEVOFF% 3-23
libraries 1-4
LKM% 3-33
load (file) 2-4
load files 2-3

Loading 67XX Driver as Instrument Module 2-4
 LVC% 3-23
 LVL1# 3-27
 LVL2# 3-27
 LVO% 3-23
 LWMAKE 2-17
 LWMAKE Utility 1-4

M

MAM% 3-17
 marker selection 3-34
 measurement functions 3-16
 MFM% 3-17
 Microsoft C 1-4, 2-17
 MKSEL% 3-35
 MODS% 3-37

N

No. of Freq 3-13
 NUMST% 3-11, 3-15
 NUMST% 3-7
 NUMST% 3-27

O

OPT% 3-25
 output leveling 3-22
 output power 3-24
 output pwr sweep 3-26
 Overview for LabWindows Users 1-4
 Overview for Non-LabWindows Users 1-5

P

pc.level 3-22
 pc.power 3-24
 pc.pwrs 3-26
 PLR% 3-43
 PM% 3-17
 PMR# 3-17
 PP# 3-37
 PROGRAM window 2-3, 2-20
 pulse modulation 3-36
 PW# 3-37
 PWRL# 3-25
 PWRMTR: EXG% 3-23

Q

QuickBASIC 1-4, 2-17

R

recall setup 3-38
 REG% 3-39, 3-41
 Requirements 1-3
 RFDR% 3-43
 RFFS% 3-43
 RTN# 3-8

S

save setup 3-40
 SENS# AM 3-29
 SENS# FM 3-33
 setam 3-28
 setcw 3-30
 setfm 3-32
 setmk 3-34
 setpm 3-36
 Source Address 2-9
 source frequency def 3-20
 special sweep 3-12
 sr.recall 3-38
 sr.save 3-40
 start-stop sweep 3-14
 SWEEP% 3-7, 3-11, 3-15
 SWEPT# 3-7, 3-11, 3-15
 SWRANGE% 3-15

T

TRIGG% 3-7, 3-11, 3-15
 Tutorial 2-9

U

ut 3-42
 utilities 3-42

W

W6700.close 3-44 - 3-45

X

XDGP% 3-37
 XPG% 3-37