VOL. 9 NO. 4 JULY/AUGUST 1982

Design of Automatic Receiver Test Systems

WATKINS-JOHNSON COMPANY

Tech-notes

Introduction

Presently, the largest utilization of automatic test equipment (ATE) is for printed circuit board troubleshooting, but after the boards are assembled into a mainframe the final test process is a time-consuming effort. The use of ATE is now expanding from the trouble-shooting of printed circuit boards to the testing of completed units or systems. Using automatic test systems to do receiver specification testing is a logical progression of this expansion of ATE use. Specification testing lends itself ideally to automation because of its highly repetitious nature. The nation-wide attempts to increase productivity are pushing the use of automation throughout industry. More specifically, the increased complexity of electronic assemblies combined with the shortage of skilled technical manpower is increasing the need for automatic test systems at all levels of testing. The development of IEEE standard 488, a standardized instrumentation bus, and the instrument manufacturers response to it, has led to the manufacture of hundreds of instruments for use on the bus. These events have helped to meet the need for automation by allowing automatic test systems to be implemented in a cost effective manner.

The measurement of any given receiver specification is basically a five-step process: 1) Connection of the appropriate instrumentation; 2) Set-up of the stimulus instruments; 3) Set-up of the receiver under test; 4) Set-up and reading of the measurement instruments; and, 5) Interpretation of the data.

During manual testing, each of these five steps is performed by a technician with the arrangement shown in Figure 1. The technician sets-up and reads the instrumentation and the receiver, then either directly records the resultant reading or uses that reading to calculate the desired data. The user's problem is that although testing of this type is repetitious, it is still necessary to utilize skilled personnel to obtain good test results manually.

Automation of the manual testing process does not alter the five basic test

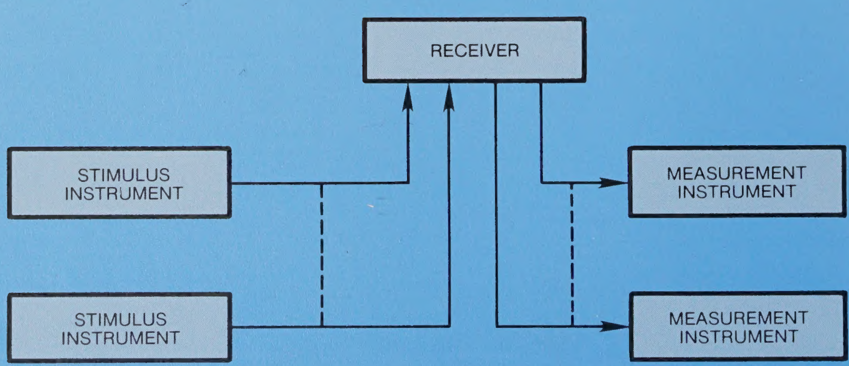


Figure 1. Basic manual test set-up.

steps. As shown in Figure 2, by allowing the system controller access to the instrumentation and receiver, it can now perform these steps automatically. Now, the connection, set-up and reading of the instrumentation, as well as the data interpretation, are all done by the system controller.

Benefits

The first and most obvious benefit of automatic testing is increased speed and throughput. Use of automatic test systems for receiver specification testing can result in an approximate 90-95% cut in test time. One important benefit derived from this increase in speed is a reduction in labor costs, which mainly occur for two reasons:
1) With increased throughput, test personnel needs decline, along with the amount of necessary test equipment.
2) Less skilled personnel can be utilized for repetitious testing, since the testing process has been reduced to machine operation. These benefits greatly ease skilled technical manpower requirements and free skilled technicians for more useful and challenging assignments, an important consideration with the current shortage of available technical manpower.

Another benefit of automated testing is that of accuracy and repeatability in the test process. The system controller has full power over the measurements; therefore, no human error is introduced into the readings. The system controller, being a machine, feels no pressure to increase output by allowing marginal units to pass through the system. In addition, the software compensates for predictable, repeatable system errors, such as cabling losses and other factors inherent in the system. There are no variations in how the tests are run from unit to unit, so there is a constant repeatability factor which provides much greater confidence in the test results.

Past Problems With Automation

In the past there have been several problems associated with automating the receiver test process, the most expensive being the very large software effort.

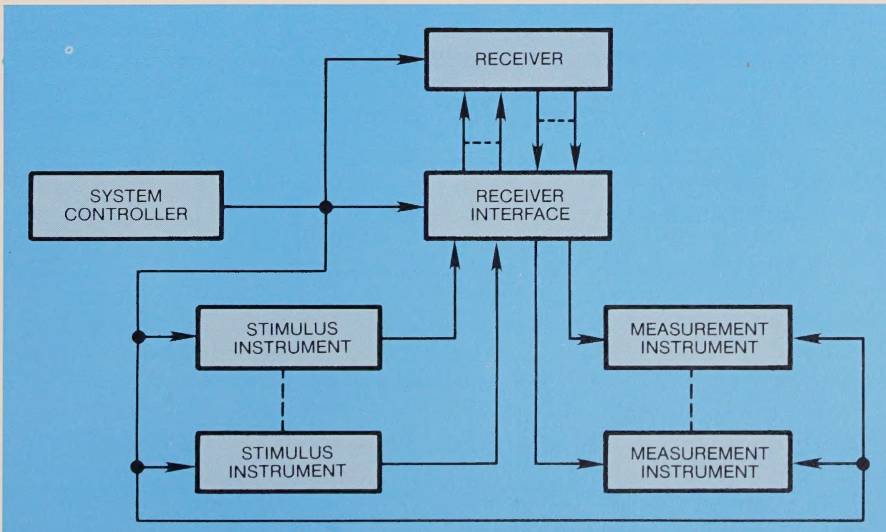


Figure 2. Basic automatic test set-up.

Programming is time-consuming, labor-intensive and, therefore, very costly. The real problem here is that every user is producing the same type of general receiver tests to accommodate their test hardware and a particular receiver. Often, the same user will duplicate their own efforts to adapt new software to another receiver, even if the new receiver is to be tested on the same test system. The long programming, and often reprogramming effort is not only costly, but because of the time involved, also leaves an interim period where manual testing of the receiver is necessary while waiting for the test software to become available and usable. An additional problem has been that programming on the system would preclude the possibility of testing on the system. Therefore, the programs that had been completed could not be used as long as the system was being utilized to accomplish software development.

The Compute-Intensive Concept

Watkins-Johnson Company has developed a line of Compute Intensive Test Equipment (CITE). The compute-intensive concept solves the past problems associated with automated receiver testing. Since software is the user's largest expense in bringing an automatic system on line, the major CITE design criteria was to minimize the user's software development. By using a software-intensive system organization, a general, highly flexible system is obtained. Menu driver software is used to guide the user through the entire programming and operation phases of receiver testing. The user, therefore, does not do actual receiver test programming, but only fills in the blanks, via the menus, to give the necessary input to the already existing software. The system software then inserts the user inputs into the resident receiver test software, which assembles the necessary test programs automatically.

Use of the powerful UNIX™* operating system helps to solve other problems previously associated with automatic receiver testing. Being a multi-user operating system, UNIX™ allows the system hardware to be configured as shown in Figure 3. With a single terminal used to control the test execution, other terminals can be connected to the CPU for use as programming stations. This allows several programmers to proceed with software development, or other tasks, simultaneously with test execution.

System Development

Development of a receiver test system is an integrated process of both hardware and software design. Generally, the hardware must at least be defined before the software development can be started. Once that hardware definition

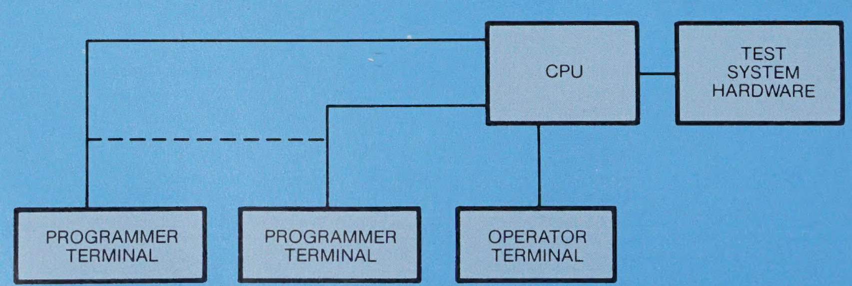


Figure 3. Multi-user ATE configuration.

*Registered trade mark of Bell Systems Laboratories.

takes place, hardware and software development can both proceed hand-in-hand. Integration of hardware and software is continual through the entire system engineering process.

Hardware Development

The heart of the receiver test system hardware is the receiver interface. This unit is responsible for all switching and interconnection between the receiver under test and the test system. The unit must be designed keeping in mind all the necessary connection possibilities between the receiver and the system instrumentation. Instrumentation should be chosen, whenever possible, with an IEEE 488 interface to simplify system integration. The test hardware and instrumentation then has only one interface point, the IEEE 488 bus, with the main system controller complex. This controller complex will generally consist of a CPU, mass storage (such as a floppy or hard disc drive), a terminal and a system printer. Figure 4 is a basic block diagram of the test system hardware. The utilization of the IEEE 488 bus increases the modularity of the system and facilitates any future expansion of the system which may become necessary.

Software Development

The first step in the development of test system software is the generation of driver subroutines for each instrument in the test system. This facilitates ease of instrument handling by the actual test programs and eliminates repetitious use of instrument programming statements. The test programs can then access the instruments simply by passing parameters back and forth, to or from, the subroutines. The use of intelligent instrumentation in the system can function as a form of distributed processing. This technique allows some of the controller workload to be off-loaded to the instrumentation. For example, an intelligent digital voltmeter with peak storage capability can take many readings, always keeping in memory the highest and lowest ones.

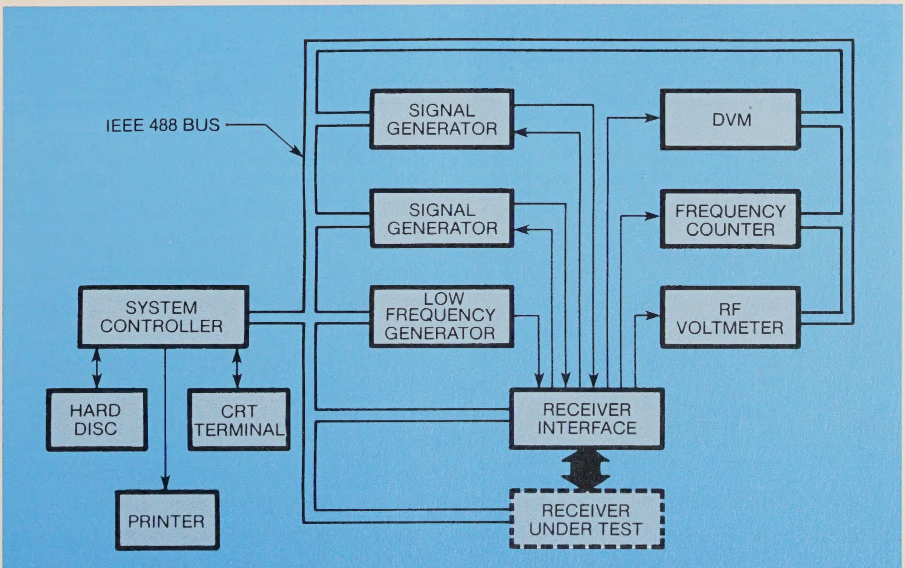


Figure 4. General automatic receiver test system.

These peak readings are then passed back to the controller. This relieves the controller of the task of reading every measurement and sorting out the highest and lowest readings.

After the generation of driver subroutines for the instrumentation, the actual receiver test programs are then generated. Since this effort becomes the most costly aspect of system generation, this is where the CITE philosophy attempts to simplify the user's effort. General purpose menu software is generated which, when operated, prompts the user to input the necessary information (see Figure 5). This then allows the user to call up the menus for the desired tests and to input the necessary parameters. The internal software then generates the necessary test programs automatically, eliminating a time consuming programming effort for the system user.

INTERNALLY GENERATED
SPURIOUS SIGNALS

0) EXIT TO TOP LEVEL

1) ENTER EXHAUSTIVE TEST FREQUENCY RANGE(S)

2) ENTER SELECTIVE FREQUENCY SET(S)

3) ENTER 'LEARNSPUR' FREQUENCY RANGE(S)

4) ENTER ACCEPTANCE PARAMETERS

5) ENTER OPERATOR MESSAGES

6) ENTER OUTPUT FORMATTING OPTIONS

SELECTION: ________________

Figure 5. Sample CITE test program menu.

The compute-intensive concept is to fully utilize the computing power of the CPU. Whenever possible, capabilities are moved from hardware to software, particularly when this move can eliminate certain pieces of hardware thereby minimizing the total system cost. As an example of this philosophy, the following discussion of noise figure will show how the compute-intensive concept can eliminate the need for a noise figure meter and provide more accurate noise figure readings utilizing an RF voltmeter which already exists in the system.

Noise figure is a measure of the amount of signal-to-noise ratio (S/N) degradation through a system. Therefore, noise figure (F) is:

$$F = \frac{S_i/N_i}{S_o/N_o} = \frac{S_i}{S_o} \times \frac{N_o}{N_i}$$

Where: S_i = Input signal power
S_o = Output signal power
N_i = Input noise power
N_o = Output noise power

Since $\frac{S_o}{S_i}$ represents the gain (G) of the system, this then becomes :

$$F = \frac{1}{G} \times \frac{N_o}{N_i} = \frac{N_o}{N_i\,(G)}$$

The output noise (N_o) has two components: First, the input noise (N_i) times the gain of the system (N_iG) and secondly, any extra noise, (N_x), which is added by the system. The noise figure equation then becomes:

$$F = \frac{N_i(G)+N_x}{N_i(G)} = 1 + \frac{N_x}{N_i(G)}$$

It is this extra noise (N_x) component which determines the signal-to-noise degradation and, therefore, the noise figure. In a theoretically noiseless system, N_x would be 0, which results in a noise figure of 1. It is more commonly acceptable to express noise figure in dB, which is simply:

$$\text{Fdb} = 10 \log F = 10 \log \left(1 + \frac{N_x}{N_i(G)}\right)$$

Our theoretically noiseless system now has a noise figure in dB of 10 log 1 or 0 dB.

To gain further insight into the problem of directly measuring noise figure, recall the previous equation:

$$F = \frac{N_o}{N_i(G)}$$

The input noise (N_i) is actually the thermal noise from the input termination, which is equal to k T_o B, where:

k = Boltzman's constant (1.38×10^{-23})
T_o = Input Termination Temperature (Room Temperature, 290°k)
B = Bandwidth over which the noise power is measured

The equation above then becomes:

$$F = \frac{N_o}{kT_oBG}$$

The practical difficulties in directly measuring the input noise (kT_oB) accurately, forces the use of an alternative method rather than a direct measurement technique. This alternative method is illustrated in Figure 6. In addition to measuring the output noise, N_1, as shown in Figure 6a, another measurement is taken of the output noise, N_2, which includes extra input noise injected by a noise source. These two measurements alone, N_1 and N_2, are all that is necessary to calculate the system noise figure.

To understand this concept, it is first necessary to define the three noise components: input noise, receiver added noise and the extra noise from the noise source. The input noise has already been defined as kT_oB. This factor times the system gain (kT_oBG) identifies the input noise component at the output of the system. In like manner, the extra noise component from the noise source can be defined as $k(T_2 - T_o)BG$, where T_2 is the equivalent noise temperature of the noise source when turned on.

The remaining noise component, that noise added by the receiver, can be derived as follows: The extra noise (N_x)

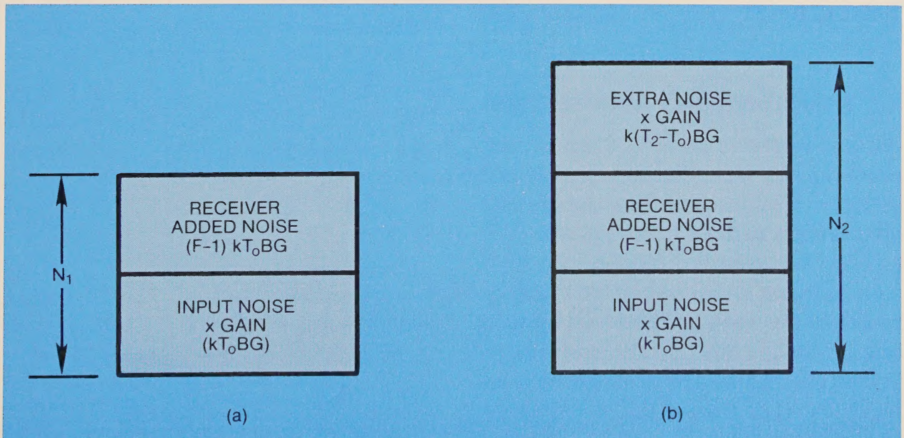


Figure 6. Noise figure measurement.

added by the receiver is equivalent to the noise output (N_o) minus the input noise, or;

$$N_x = N_o - kT_oBG \tag{1}$$

Solving the previous noise figure equation for output noise yields the following:

$$F = \frac{N_o}{kT_oBG}$$

$$N_o = kT_oBGF \tag{2}$$

Substituting equation (2) into equation (1) will yield the formula for the noise added by the receiver (N_x).

$$N_x = kT_oBGF - kT_oBG$$
$$N_x = (F-1)\,kT_oBG$$

It is now possible to define the ratio N_2/N_1 as follows:

$$\frac{N_2}{N_1} = \frac{\text{Input Noise + Receiver Added Noise + Noise Source Noise}}{\text{Input Noise + Receiver Added Noise}}$$

$$\frac{N_2}{N_1} = \frac{kT_oBG + (F-1)kT_oBG + k(T_2 - T_o)BG}{kT_oBG + (F-1)kT_oBG}$$

Simplifying:

$$\frac{N_2}{N_1} = \frac{FT_o + (T_2 - T_o)}{FT_o}$$

$$= 1 + \frac{(T_2 - T_o)}{FT_o}$$

Solving for F:

$$F = \frac{\dfrac{(T_2 - T_o)}{T_o}}{\left(\dfrac{N_2}{N_1} - 1\right)}$$

Converting to dB:

$$FdB = 10 \log \frac{(T_2 - T_o)}{T_o} - 10 \log \left(\frac{N_2}{N_1} - 1\right)$$

The term $10 \log \frac{(T_2 - T_o)}{T_o}$ is a measure of the noise power from the noise source. This is defined as the excess noise ratio (ENR) specified by the manufacturer of the noise source and will typically vary a few tenths of a dB over the frequency range of use. For most noise diodes available, the ENR will be approximately 15.2 dB. The final noise figure equation then becomes:

$$\mathrm{FdB} = \mathrm{ENR} - 10 \log \left(\frac{N_2}{N_1} - 1 \right)$$

The only two unknowns, N_2 and N_1, can be read directly at the output and fed into this equation.

This concept more fully utilizes the computing power of the CPU to eliminate the need for a special piece of hardware. This philosophy, when carried throughout each individual test, will produce an efficient system of minimal hardware with more fully utilized software capabilities.

Receiver Diagnostics

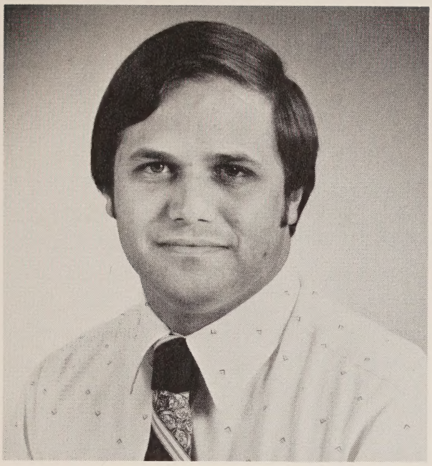
After all the receiver's specifications are gathered and analyzed, the most obvious question then becomes, "Can the troubleshooting process be automated if the specifications are out of tolerance?" There are two ways to approach the automated diagnostics process. The most obvious is to mechanically interface to the internal circuitry of the receiver and trace the signal through the receiver to be certain of proper signal processing. There are numerous problems associated with this method. First, depending on the mechanical design of the receiver, it may require extensive engineering to solve this interface problem. Second, interfacing to the required number of test points may load the receiver too heavily and cause its operation to cease. If these problems can be overcome, it may be cost effective to pursue this option in a production environment for a single receiver product, but in a depot situation, where many different receivers must be maintained, the cost of developing many interfaces of this kind cannot be justified.

In reality, even in a production environment, it generally is not cost effective to use this method of automated diagnostic testing. The technician working on a particular receiver in production becomes intimately familiar with that product. He can then look at the resulting specifications and identify the problem area from experience. Therefore, diagnostic testing to a module level, generally is never cost effective in a production environment. In contrast, the depot technician works on many different receivers and never really attains detailed familiarity with all of them. His need for diagnostic testing, therefore, is much greater than that of the production technician's. The solution to this problem is once again the compute-intensive concept. Software analysis of the resultant specifications can be used to identify the failing module. In actuality, this automates the production technician's thought processes in arriving at the module with the highest probability of failure. This process, while not true diagnostic testing, provides a solution to the problem which is usable and economically feasible.

Management Enhancements

Due to the increased power of smaller computers available for use in these test systems, it is possible to employ data management techniques from the data processing industry within the testing environment. As testing takes place, individual test data can be stored within the printer controller. At the completion of testing for any particular receiver, the printer controller can first scan all of the data collected for that receiver. If all data is not within specification, a failure list can be printed for the receiver. If the data is found to be acceptable, the printer controller can begin printing a customer-deliverable data sheet which can be formatted as desired. The next receiver could then be started as the first receiver's data sheet is printing. The data for receiver number two could be loaded under interrupt control. This method of distributed processing is very useful for increasing overall throughput. As test data is collected by the system, it is stored and kept available for management use. This process then forms a complete product data base. Applications software can then be used to analyze this data base to obtain product trends or other useful management information as necessary. It would be possible to retrieve a particular piece of data from any particular receiver, such as noise figure for serial number 87, for example, or to analyze data from a group of receivers, such as the average noise figure of serial numbers 20-100. This becomes a valuable tool for product analysis and enables the user to see any trends which may develop during the life of a product.

Author:



Scott A. Moser

Mr. Moser is currently an Applications Engineer for the ATE systems product area in the SSE Division.

He began his work at Watkins-Johnson, in the CEI Division, where he worked on the design of the WJ-8718 HF Receiver. After completion of the WJ-8718 development, he was tasked with transitioning the receiver from engineering into production. Part of the responsibilities for this transition involved the development of an ATE system for final receiver specification test of the WJ-8718. Mr. Moser was also involved in the design of several digital control options for the WJ-8718.

He was later responsible for establishing complete ATE support for the production of the WJ-8610 series of VHF-UHF receivers. This was accomplished by expanding the Company's ADATE product line to do receiver testing, which involved rewriting parts of the ADATE operating system software.

Mr. Moser holds a B.S. degree from Capitol Institute of Technology and is currently pursuing an M.S. degree in Administration and Management at Hood College in Frederick, Maryland.

WATKINS-JOHNSON COMPANY
3333 HILLVIEW AVENUE
PALO ALTO, CA 94304
(415) 493-4141

BULK RATE
U.S. POSTAGE
PAID
PERMIT NUMBER
317
SUNNYVALE
CALIFORNIA

WATKINS-JOHNSON

Manufacturing and Office Locations

United States

SALES OFFICES

CALIFORNIA
Watkins-Johnson
2525 North First Street
San Jose, 95131
Telephone: (408) 262-1411

Watkins-Johnson
440 Kings Village Road
Scotts Valley, 95066
Telephone: (408) 438-2100

Watkins-Johnson
831 South Douglas Street
Suite 131
El Segundo, 90245
Telephone: (213) 640-1980

DISTRICT OF COLUMBIA
Watkins-Johnson
700 Quince Orchard Road
Gaithersburg, MD 20878
Telephone: (301) 948-7550

GEORGIA
Watkins-Johnson
4250 Perimeter Park, South
Suite 123
Atlanta, 30341
Telephone: (404) 458-9907

MARYLAND
Watkins-Johnson
700 Quince Orchard Road
Gaithersburg, 20878
Telephone: (301) 948-7550

MASSACHUSETTS
Watkins-Johnson
5 Militia Drive
Suite 11
Lexington, 02173
Telephone: (617) 861-1580

OHIO
Watkins-Johnson
2500 National Road
Suite 200
Fairborn, 45324
Telephone: (513) 426-8303

TEXAS
Watkins-Johnson
9216 Markville Drive
Dallas, 75243
Telephone: (214) 234-5396

International

ITALY
Watkins-Johnson
S.p.A.
Piazza G. Marconi, 25
00144 Roma-EUR
Telephone: 59 45 54
Telex: 612278
Cable: WJ ROM I

UNITED KINGDOM
Watkins-Johnson
Dedworth Road
Oakley Green
Windsor, Berkshire SL4 4LH
Telephone: (07535) 69241
Telex: 847578
Cable: WJUKW-WINDSOR

GERMANY, FEDERAL REPUBLIC OF
Watkins-Johnson
Manzingerweg 7
8000 Muenchen 60
Telephone: (089) 836011
Telex: 529401
Cable: WJDBM-MUENCHEN

Watkins-Johnson
Burgstrasse 31
5300 Bonn 2
Telephone: 35 30 91
35 30 92
Telex: (886) 9522

The Watkins-Johnson *Tech-notes* is a bi-monthly periodical circulated to educational institutions, engineers, managers of companies or government agencies, and technicians. Individuals may receive issues of *Tech-notes* by sending their subscription request on company letterhead, stating position and nature of business to the Editor, *Tech-notes*, Palo Alto, California. Permission to reprint articles may also be obtained by writing the Editor.

Printed in U.S.A.