LA-13027-M
Manual

UC-706
Issued: September 1995

LACHESIS - An Instrumentation System for Obtaining Containment and Environmental Data

Robert G. Deupree
F. Ross Oblad

Los Alamos
NATIONAL LABORATORY
Los Alamos, New Mexico 87545

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

LACHESIS - An Instrumentation System for Obtaining Containment and Environmental Data

by

Robert G. Deupree and F. Ross Oblad

ABSTRACT

The instrumentation system developed to obtain containment and environmental data in the LYNER complex is presented. The primary purpose of this report is to familiarize potential operators of the system with the details of its use.

The instrumentation system has three major hardware modules - 1) the sensor power source, amplifier, and signal conditioner module, 2) the digitizers, and 3) the computer controller. Each of these is described with emphasis on the steps required to make that component perform effectively. In addition the roles of activities of other people besides the Los Alamos shot engineer who are required to ensure the success of the system are outlined.

I. INTRODUCTION

As Los Alamos began to study the feasibility of the LYNER complex, it became apparent that our then current system for collecting containment and other related relatively long time scale (a few seconds) information was not ideally suited for our expected needs. Thus, once construction of LYNER was begun, we developed the requirements for an instrumentation system and to investigate the state of current instrumentation technology. Based on electronic trade shows, standard vendor demonstrations, and specialized demonstrations we asked vendors to perform, we selected Hewlett-Packard (HP) to supply the digitizers and the controlling computer system and Pacific Instruments (PI) to supply the signal conditioning, amplifier, and power supply modules. We should emphasize that our decision was based on our situation and commercially available products at a time of rapidly changing technology, and that our choice should not be interpreted outside these parameters.

We shall begin with an overview of the complete system. Then we shall divide the system into its components and discuss these individually. Finally, we shall discuss the steps that must be carried out to insure successful data acquisition.

II. OVERVIEW

The LACHESIS (Laterally Archiving Containment, Health, Environment, and Safety Instrumentation System) is composed of a number of identical subsystems which may be placed in a variety of locations, a HP 725 workstation located in the Monitor Room in CP-1 at the Nevada Test Site (NTS), and a HP 735 workstation in Los Alamos. Each subsystem contains a HP V382 workstation (also known as a slot zero controller) mounted in a VXI case. All workstations use the UNIX operating system and are connected to each other using Ethernet and UNIX protocols. However, any subsystem may be operated alone without any external communication.

A subsystem is composed of the slot zero controller, the amplifier and signal conditioning modules, and analog to digital (A/D) boards. The slot zero controller includes a monitor, keyboard, mouse, hard disk, and DAT tape drive external to the VXI case. Each subsystem is identified by a name. The current subsystem names and locations are: Sparta in the DX-12 alcove near U1a, Athens and Thebes in the East Users Alcove near U1g, Mycenae in CP-45, and Pylos and Ithaca in Los Alamos.

The Monitor Room workstation is called lachesis (lower case to distinguish it from the name of the system as a whole), and the workstation in Los Alamos is called deupree. Two important features to note are that a user can log onto any workstation from any other workstation, and that data can be transferred remotely from any workstation to any other easily (and even without any user assistance).

The tasks of the lachesis workstation are to act as a source of remote control for all the subsystems, as a collection and storage point for all data, and as a tool for data presentation and limited data analysis. The deupree workstation, in addition to allowing direct communication to all other workstations from Los Alamos, is expected to perform the final data reduction, storage, and analysis. Both lachesis and deupree have a signal processing and quality data presentation package, MATLAB (1,2) for data analysis and presentation.

The software for LACHESIS has been written in a variety of computer languages. All subsystems contain programs written in VEE, a HP iconic programmer (3,4,5), in C, and in UNIX scripts. All user access to data acquisition is through VEE programs, which in turn access the C programs and UNIX scripts. The C programs communicate directly with the A/D boards over the VXI bus, while the UNIX scripts primarily control file manipulation and interactions among the various workstations. The lachesis and deupree workstations contain MATLAB scripts written in a language very much like C.

III. SUBSYSTEM COMPONENTS

We now turn to a discussion of the individual subsystem components: the amplifiers/signal conditioners, the A/D boards, and the slot zero controllers. The first two will be described in some detail, because understanding many of the tasks we perform will require user knowledge on a fine scale. Complete information is provided for the amplifier and signal conditioning modules in (6), and for the A/D boards in (7).

A. Amplifier and Signal Conditioning Modules

As we began to narrow the vendor candidates for the A/D boards, it became apparent that we would require more protection for the digitizers than vendors usually supply. Most of the reason comes from the fact that nuclear explosions can generate relatively large, if short lived, voltages in cables. These voltages can far exceed the 16 volt common mode and the 42 volt peak that the HP digitizers, for example, can receive without damage. Thus, we concluded that external signal conditioning was required.

One of the most desirable features for signal conditioners and amplifiers was the ability to remotely program them. This is driven by the fact that the LYNER complex does not guarantee either quick or easy access. Delays of an hour or more to get underground are not impossible. Furthermore, health and safety require limitations on the number of people underground at one time, so that there is the possibility that personnel may not be able to get to the equipment even if there is a need to do so. Being able to access all equipment from the Monitor Room therefore provides clear advantages. The activities one might desire to perform remotely include measurement of excitation voltage and current, the ability to disable excitation, the grounding of amplifier input, shunt calibration, voltage substitution calibration, and the measurement of amplifier output.

Another key concern was to have independent and isolated channel units. During the destructive tests we envisioned, some sensors are destroyed and cables may short causing transients on adjacent channels or rendering excitation useless for other surviving channels that use a common excitation source. In order to obtain maximum adjacent channel isolation, amplifier systems must include independent and isolated sensor excitation and amplifier power supplies as well as physical modularized independence. This type of architecture generally yields maximum adjacent channel isolation. We therefore required isolation and independence out to 120 V ac.

Explosive testing can produce electromagnetic pulses with substantial amplitude. Additionally, other diagnostic equipment uses switched high voltage and current in their operation causing electromagnetic interference. All of these external sources of energy are introduced to our balanced measurement system as common mode voltages. To protect our data and our equipment from these pulses, we imposed a constraint of 300 volts common mode rejection on the signal conditioners.

Since we could not define the suite of measurements we would be called upon to make in the LYNER complex, we needed flexibility in both signal conditioning and amplification to accommodate the widest possible range of sensors. Primary considerations included programmable amplifier gains between 1 and 1000, programmable excitation type (voltage or current) with programmable amplitudes, and programmable low pass filtering.

A final consideration was channel density. Space underground was limited, so that the number of signal conditioners that could be accommodated in a rack was a significant criterion.

After a market study was performed, we selected Pacific Instruments, Inc. to supply the amplifier and signal conditioning modules. These signal conditioning amplifier modules we selected have the capability to program a number of features, including the gain, the type of excitation and excitation level, and the filter cutoff frequency. In addition, one may also enable or disable the autozero and autobalance features.

We chose the gain for each sensor so that the A/D board is presented with an input voltage between -10 and +10 volts. The A/D board can accept voltages in the range -16 to +16 volts as input, so we have some margin of safety. This margin is further enforced by the fact that the amplifier saturates at about 13 volts (both positive and negative). Thus, the A/D board should be protected by the amplifier. The user must then choose the gain so that the expected variation of the sensor output during a test is sure to be within the -10 to +10 volt range. The fact that the A/D boards currently seem to be limited by the resolution of the 16 bit A/D converter allows the user to error on the low side with less risk that the data obtained will be swamped by the noise level of the system.

However, we must recognize that the signal conditioning amplifier itself adds noise to the measurement. At large gains this can become significant. Note that the amplifier noise specifications are "referred to input". For example, in the 0.1-10000 Hz frequency range, the amplifier noise specification is 2 μV referred to input and 300 μV referred to output. At a gain of 10, the actual noise is 320 μV and 500 μV at a gain of 100. This is becoming significant with respect to the 488 μV resolution of the 16 bit digitizer for the full 16 Volt range, and is even more so if a more restricted voltage range of the A/D board is used (see the next section). Clearly, some thought must be given in the determination of the gain selected.

The signal conditioning amplifier also allows the user to specify whether the excitation is constant voltage or constant current. The sensor determines which type is required. The amount of voltage or current excitation is also programmable. If the selection is constant voltage, one may also choose to utilize remote sensing. For the price of an extra cable pair from the amplifier to the sensor, the amplifier can sense the voltage at the sensor and adjust the excitation voltage so that the input voltage at the sensor is the required amount. However, if the sense leads should become open, incorrect voltage may be sent down the cable and the sensor readings become erroneous. Our tendency has been to use this remote sensing feature except when the cables are very short, there are insufficient pairs in the cable to allow the sensing, or the sensor has its own internal voltage regulator.

When we purchased the amplifiers, we had to make a decision on the type of filtering we wished. We chose a four pole Bessel low pass filter for all amplifiers. The allowed cutoff frequencies are 10, 30, 100, 300, 1000, 3000, 10000, and 30000 Hz. In addition, the filter may be bypassed entirely. We have adopted a relatively conservative stance that the cutoff frequency should be 10 times higher than the highest frequency of interest in the signal. However, if the channel has high level noise sources, then this value of 10 may have to be compromised. We also note that the Bessel filter imposes an increasing phase shift in the higher frequency components which result in a time delay in the response of these components through the filter.

The autozero feature allows the amplifier to zero each time a change in the gain or cutoff frequency is made or when the amplifier is turned on. The amplifiers are generally stable enough that this initial zero is sufficient to sustain a true zero condition thereby eliminating a system introduced zero shift on the recorded data. The major drawback of this feature is a periodic bit shifting on the digital to analog converter used to set the zero which results in a 500 to 1000 μV zero shift on the output of the amplifier. This shows up in the data as an instantaneous shift in the value of the physical variable. Autozero will not prevent a drifting sensor or amplifier from causing a shift of zero in the data. We are investigating techniques to force a zeroing of the amplifier just before taking zero time data to compensate for a drifting amplifier. Amplifier induced zero shifted data is most noticeable when the power to the amplifiers has been interrupted thereby forcing a zero when the power is restored. A test sequence is also under investigation to identify drifting amplifiers. We have enabled autozero for all channels.

The autobalance feature allows another digital to analog converter to force the amplifier to zero regardless of the signal input. However, the feature is range limited and thus will not balance a large offset. If the zero of the sensors drift, the autobalance may have to be relatively frequently invoked. Although this feature is useful in eliminating small zero shifts, we have elected not to enable it.

The signal conditioning amplifier provides two modes of calibration. Voltage substitution via a connector on each rack of ten amplifiers and a simulated shunt calibration for bridge resistive type transducers. Both modes are programmable in execution while the simulated shunt calibration can be programmed in selected voltage or physical variable levels. These features are not as yet implemented by LACHESIS. A more involved calibration scheme is planned for future addition.

Features supporting remote testing of the analog portion of LACHESIS are present in the amplifiers. Transducers can be tested by measuring excitation voltages and currents. These measurements can be displayed on the front panel or read by the digitizer. System noise measurements can be made by disabling excitation while looking at the amplifier output. The amplifier input can be grounded, allowing amplifier noise to be measured.

These features have been valuable in diagnosing amplifier and transducer failures during deployment and operation of LACHESIS. Input and output test jacks on the front panel have also facilitated installation and testing.

All of these features may be controlled from either the front panel of the amplifier or remotely through either an RS-232 port or a GP-IB port. Unfortunately, the software for remote control supplied by PI requires a personal computer with Windows. We are currently writing software which we can use from our UNIX based slot zero controllers.

Points of concern pertaining to the amplifiers include loss of programming during a power shutdown, zero drift, and failures of input balance. Underground power outages are more frequent than desirable. After these outages 1 or 2 amplifiers have lost their programming in that gain is reset to 1, excitation to 10 volts, and filter set to wideband. Any test or calibration settings are returned to their default conditions during a power outage.

Zero drift as mentioned previously is usually discovered by looking at the data. Sudden shifts in level at the time of a power outage is the noted condition. If a zero drift is suspected the testing method is to force an autozero by changing the gain to the same value while observing amplifier output. Shifts greater than a few millivolts indicate a drifting amplifier. Since LACHESIS is usually in archiving mode, this long term drift is a concern.

Failure of the amplifier input circuitry causing an unbalanced condition is another concern. The observed failure is an unexplained large shift in data value. During this failure mode amplifier tests of input level, output level, excitation voltage and current, impact of grounding the amplifier input, and impact of disabling excitation have been normal, not showing the failure. Only amplifier substitution has been successful. This condition has only been observed with resistive bridge transducers.

When these or other points of concern have been identified in an amplifier, it has been sent to PI for repair. Investigation is underway to design routine testing procedures and build them into the software that will identify and flag amplifiers that have failed in these or other modes and if possible will correct the problem so that data can be accurately acquired without changing the amplifier.

B. A/D Boards

The HP E1413 A/D boards take one slot each in the VXI case. We have adopted the convention that the first board on a subsystem has the logical address of 24 and the second board the logical address of 25. Currently, the software allows only two E1413 boards on any given subsystem.

The A/D boards allow up to 64 channels with a single 16 bit A/D converter. The HP convention, which we adopt, is to number the channels from 100 to 163. Each channel is scanned in a sequence provided by a user generated scan list. The channels can be scanned no faster than the minimum measurement window of 10 μs, but we have discovered some problems when operating at this limit and prefer to sample at speeds slower than 13 μs. This generally presents no difficulties for the types of measurements we make.

Each A/D board has approximately 64,000 words of on board storage. The board offers the option of filling up the memory and then terminating data collection or recovering the data from memory while data collection continues and then overwriting the already read memory with new data. Data extraction can be sufficiently rapid that no data are lost. For our initial development phase, we have chosen not to overwrite data in memory so that it could all be recovered in case of failures elsewhere in the system. Thus, the upper limit to the time for which we can collect data is determined by the amount of memory on the boards.

The flexibility in data collection is provided by the scan list, which may be repeated either a set number of times or continuously until the memory is filled up. The parameters the user must specify in addition to the scan list are the measurement window time interval or time between sampling adjacent entries in the scan list (Δt), the time interval between consecutive beginnings of the scan list (ΔT), and the number of times the scan list will be executed (N). For execution of the scan list until all of on board memory has been filled, set N=0. The user must allow 30 μs + Δt between the end of one scan and the beginning of another. The scan list may contain no more that 1024 entries.

One of the most important tasks the user must perform is to generate the scan list. The user must balance the desired sampling rates for all channels, the total number of boards, the total time duration desired for data collection on each board, and the memory available. Some extra degree of freedom may be provided by channel 100, which has been decreed to be a dummy channel. On the KISMET test we decided to field most channels at a 1 ms sample rate, with the remaining approximately ten channels (related to zero room activity) sampled at 250 μs. This had the advantage of producing a relatively simple scan list with the most frequently sampled channels appearing four times and all other channels once, but it sampled many channels in remote parts of the tunnel more frequently than required and did not allow us to collect data for the zero room measurements as long as we would have liked (in retrospect). One might think that the solution would be to increase the ratio between the sample time of the least frequently sampled data to the sample time of the most frequently sampled data. However, a complication arises in the need to sample channels between the recurrences in the scan list of the most frequently sampled channels. This may require more channels than are available, either sampling some channels more frequently than desired or using a dummy channel (which uses memory with no useful information).

An example might help to clarify this. Suppose that we have three channels (say, 101-103) that we wish to sample every 100 μs and four more (104-107) that we wish to sample every 200 μs. Then the three channels must appear twice in the scan list for every appearance of each of the other four channels. We can see that the following scan list with a 20 μs sampling interval will satisfy these requirements (we note that this will not satisfy the 30 μs + Δt time interval required between the end of one scan and the beginning of the next scan of the scan list, but the example is simpler this way):

101 102 103 104 105 101 102 103 106 107

However, now suppose that we want to sample the second set only every 600 μs, requiring the more frequently sampled channels to appear six times for every appearance of each appearance of the other channels. One could produce the following scan list:

101 102 103 104 101 102 103 105 101 102 103 106
101 102 103 107 101 102 103 10x 101 102 103

However, we see that we do not have enough of the channels being sampled less frequently to fill in all the required gaps - hence the fictitious channel 10x. We need to replace the channel 10x with either a real channel or a dummy channel. In either case, we may be taking data either more frequently than we would like or using up board memory by storing data for a dummy channel. It is clear that efficient use of the board memory requires some dexterity with the scan list. This is probably the most difficult part of the system to master, although simple scan lists do lead to successful, if less than ideal, data acquisition.

Additional flexibility in measurement is provided by five input voltage ranges including autoranging capability. The available ranges are ± 16 volts, 4 volts, 1 volt, 0.250 volts, and 0.0625 volts. Hewlett Packard has indicated that in some cases, the autoranging feature will not settle on a measurement during the minimum measurement window of 10 μs. We have experienced another problem with the minimum measurement window in that under certain conditions the previous channel measurement will be recorded as the current channel value. Setting the measurement window to 11 μs or greater eliminates this problem. The problem appears to be related to the scanner.

Once the system has been checked out, it is expected that it will be run continually in the long term data collection mode. A UNIX script is run on the lachesis machine which directs each subsystem active on the event to switch from this archiving mode to zero time collection mode. Since the file which contains the information is not read until a 100 data point collection log is finished, the user must send this command at least 35 minutes before zero time. Once this file is read, the VEE program turns control over to a C code which configures the A/D board and arms it to await a trigger to begin data collection. This trigger comes from an external box mounted in the rack. This box is provided by timing and firing personnel. The trigger comes into each subsystem through the front panel of the slot zero controller. Our software redirects this into one of the TTL lines on the vxi bus and indicates to all A/D boards present on the subsystem that this line is where the trigger will come. Once the zero time data has been collected and written to files, the VEE program resumes control and archiving begins again. Another command to collect zero time data can be sent without resetting the software. It should be noted that the A/D board triggers on a falling pulse. Thus the input trigger pulse must be sufficiently short so as not to include unacceptable time delays, or the pulse must be inverted so that the leading edge of the pulse spike is decreasing. The pulse supplied by timing and firing at LYNER meets the former condition.

C. Slot Zero Controller

The LACHESIS slot zero controller is an UNIX based HP workstation mounted in the first slot of the vxi chassis. This allows the controller to serve two functions - as the slot zero controller, it is plugged directly into the vxi bus and thus can communicate easily with the vxi instruments, and as a computer it can provide the software to perform all data collection functions. Furthermore, the workstation has a SCSI bus connection to an external 2 Gbyte system disk and a DAT tape drive, a GP-IB port which can be connected to the PI amplifier chassis and thus be used to communicate with the amplifiers, and two RS-232 ports for printers and other peripheral equipment.

In addition, each slot zero controller has the full array of UNIX networking capability via Ethernet, as do both the lachesis and deupree workstations. This allows us to send data and software modifications from one system to another with very little effort.

The workstations have other standard UNIX features. Each system is both multiuser and multitasking. This has particular advantages if one system is collecting data and you wish to log onto that system remotely to carry out some relatively mundane task. We should also point out that, like all other UNIX systems, you wish to avoid shutting down the machine except by the prescribed shutdown procedure. While it is relatively unlikely that files will get corrupted by just switching the machine off or removing power from it, it can happen, and the consequences could be painful if the wrong files are corrupted. To help remove this possibility, we have each slot zero controller backed up by an uninterruptible power supply (UPS), which will provide power for about thirty minutes after commercial power has been lost. This should provide enough time for either commercial power to be restored or the system to be properly shutdown (this also can be performed by logging on remotely).

Using the UNIX based privilege hierarchy, we have generated three users with different levels of privilege: test, lanl, and root. The user root has all the usual UNIX root capabilities and is currently the only user who can shut down the system. Test is the user that most people can expect to sign on as. This user may execute the system preparation and data collection software we have developed (VEE codes, C codes, and UNIX scripts), but does not have the privilege of modifying this software. This privilege is reserved for the user lanl.

One other use of the UNIX privilege system is to lock the scan and trigger lists for a test once they have been shown to fulfill the user's intentions. This is done by making these files have read only permission for all users. Once a test has been completed, these files will have to be unlocked before new scan or trigger lists can be generated. If this is not done, the user will get a VEE error stating he cannot write to those files.

Once signed on as test, a user can bring up the individual VEE codes using names in the test directory. These are actually links pointing to codes in the lanl directory tree. The entrance to VEE is provided by typing vee.

We do not propose to provide the details of the software here as this information has been provided elsewhere (8). We summarize the VEE codes that we currently have in Table 1.

The individual slot zero controllers have all been set up identically to ease the transfer of software, performing software updates, and transmission of data. Any slot zero controller can be logged onto remotely from any other slot zero controller or from the lachesis workstation in the CP-1 Monitor Room or the deupree workstation in Los Alamos. UNIX scripts and suitable login files are provided to enable this to be as transparent as possible to the user. To login remotely, merely type remote system, where system is the name of the subsystem to which you wish to log on. Further instructions are supplied on the screen after you enter this command. Remote file transfer with the UNIX rcp command is very easy. While the links to these other workstations have been made, it should be pointed out that each individual subsystem can stand alone, preparing for data collection, collecting the data, and storing the data locally. It is expected, however, that the final data reduction, analysis, and final storage will be done on either the lachesis or deupree workstations. If the network is not available, data may be transferred using DAT tapes as all workstations have DAT tape drives.

D. Connections

There are permanent cabling and connections for the subsystems that are currently underground in the LYNER complex (Sparta, Athens, and Thebes). Since the other systems are not permanently installed for continual data collection anywhere, they have only temporary cabling. For those subsystems with permanent cabling, the amplifiers and the A/D boards are connected with small diameter coaxial cables between the amplifier output and the A/D input. The amplifier end is not grounded to any chassis. The low input at the A/D input is grounded to the vxi chassis with a jumper. Each bank of eight channels is grounded together with the group ground, which is then grounded to the vxi chassis. The coaxial cables are routed together, but are kept away from the digital connecting cables used for amplifier control (the GP-IB link and the ribbon cable between racks of amplifiers).

The cabling to the amplifier input is individually shielded twisted pair cable. Typically three pair are needed for input, excitation, and excitation sensing. The shields of all three pair are connected to the guard at the amplifier end. The transducer end is connected as required by the transducer, which typically has the input shield grounded at the transducer and the excitation and sense shields floating. Often the negative excitation lead is connected to ground at the transducer. In addition to these shields for the individual twisted pair, there is an overall shield which is connected to ground at the 15 psi bulkhead between the Users Alcoves and the experiment areas. If the cable enters the zero room, the overall shield is grounded to the primary containment plug bulkhead where the cable enters the plug.

IV. STEPS FOR DATA ACQUISITION

With the successful completion of the KISMET event, we have been able to identify the steps required to ensure the acquisition of valid and useful data. Most of these steps have been formalized with checklists which are presented in the Appendix. Some of the steps are ones which we carried out informally on the KISMET test, but which should have a more formal methodology. We outline the steps as follows.

1. **Obtain Physical Measurement Requirements** - These requirements have traditionally been generated by our customers within Los Alamos, but on KISMET we began to define some of these ourselves. For our customers not in the group charged with LACHESIS fielding, these should be obtained in writing. The desired information should include the type and location of measurement, the maximum and minimum signal expected (in physical units), the data sampling requirements, the expected duration of the signal of interest, and any special considerations that may apply. These requirements should be acquired by the Los Alamos shot engineer.

2. **Create Internal Requirements Document** - Once all the measurement requests have been received, we need to collect all this information together in one table or document. For each measurement this must include the type of measurement, the expected signal range, the sample rate, the amplifier gain, the cutoff frequency, the desired time span for measurement, and the sensor model, range, and sensitivity. We have adopted the guidelines that the sampling rate should be a factor of ten larger than maximum frequency expected in the signal, and that the cutoff frequency should be no larger than a third of the sampling frequency. We do recognize that there may be some specific measurements for which these guidelines cannot be met. This task should be performed by Los Alamos personnel.

3. **Create Cable Request** - Once we know the transducers that we need, we can group these together by location and by type to see how many cables of what type are required. For example, a triaxial accelerometer package can be put together with only one 9-pair cable. Atmospheric packages which contain air pressure, air temperature, humidity, and flow can also be done with one cable. Once these have been determined, a memo must be written to the site engineering group (currently DX-14) for them to prepare the cables for our use. Care should be taken to make sure that all sensors from a given package are on the same A/D board and preferably on adjacent channels. This task should be done by the Los Alamos shot engineer.

4. **Create Sensor List** - Once we have identified the transducer types and ranges that an experiment requires, we must obtain the sensors and make sure they are calibrated. Thus a list of the sensor requirements must be sent to our Nevada Engineering support organization (currently EG&G) for them to provide the sensors from their inventory or purchase them if necessary. Time must be allowed for sensor calibration by their Calibration Lab. This is the responsibility of the Los Alamos shot engineer.

5. **Prepare Channel List, Scan List, and Trigger List** - With

all the measurements defined, we may assign each measurement to a subsystem and an A/D board on that subsystem. We must make those assignments in such a way to make sure that we can accommodate all measurements with the appropriate sample rate and total data collection time. To determine this, we must simultaneously generate the trigger list (the three quantities Δt, ΔT, and N defined in IIIB) and the scan list. Efforts should be made to keep these as simple as possible given the measurement constraints. Also keep in mind the cable constraints listed above. This is a task for Los Alamos personnel.

6. **Enter Requirements Document Information into Database** - There are 128 files on each subsystem which contain the information required to obtain and reduce data and store the results in the appropriate files (one file per A/D board channel with two A/D boards per subsystem). The combination of these files is referred to as the database. Each subsystem has its own database , and that database resides only on that subsystem. There are two VEE codes which can modify the database (see Table 1): Gen_Cal which was used on KISMET, and VeeDB, which is currently experimental but which we expect to supersede Gen_Cal in the near future. These codes can be accessed by logging on (either remotely or directly) to the subsystem which will be controlling the sensors and typing vee. After database modifications are complete, the VEE program Print_Cal can then be run and the entries checked by looking at the generated files Print_db_file and Print_db_file1. These tasks must be performed by Los Alamos personnel.

7. **Enter Sensor Calibration Information into Database** - Once the Nevada Engineering support organization has delivered the sensors and their calibration sheets to the NTS, the calibration information must be entered into the database. The same codes are used as in the previous step. Three pieces of calibration information are required: 1) a physical reference value - one of the physical values from the calibration sheet which is near the expected physical value of interest, 2) a voltage reference value, which is the voltage on the calibration sheet which corresponds to the physical reference value, and 3) a linear coefficient - the difference between two voltages on the calibration sheet divided by the difference between the two corresponding physical values on the calibration sheet. These points on the calibration sheet should be selected to be within or near the expected signal produced during the test. This task can be done by either Los Alamos or Nevada Engineering support personnel, but only after sufficient training on determining the three pieces of calibration information has been given.

8. **Installation and Check Out** - Once all the cables and sensors are available, the cables need to be terminated and the transducers wired and connected. The amplifier parameters need to be entered into the amplifier, either from the amplifier front panel or with the VeeDB software. When the sensors are connected,

the VEE codes Multiple_channel_plot and chan_plt can be used to check them out. Multiple_channel_plot is particularly useful for packages which contain more than one sensor, most notably for performing flip tests with triaxial accelerometer packages. Check the physical values to see that they make sense. Once the sensors appear to be working appropriately, run the long term data collection VEE program Archive_Plus_Zero for a few days to see that the expected variation in the physical variables is present. This is particularly useful for the atmospheric and pore pressure sensors. This step should be performed before any sensor is installed in a location from which it cannot be retrieved. Continue long term data acquisition once all sensors have been installed. These tasks may be performed by Nevada Engineering support personnel, with assistance from Los Alamos personnel as needed.

9. **Check Implementation against Database** - Once the sensors have been installed, all information needs to be checked against the database. Since the amplifiers can be set independently from the database by using the front panel, check the amplifier settings. We also need to check that the scan and trigger lists provide the sampling rates and total data acquisition timescale desired. This can be determined by collecting zero time data and checking the length of data record and the time interval between data points for each sensor. Checklists are supplied (see Appendix) to assist in this process. This should be carried out under the supervision of the Los Alamos shot engineer.

10. **Check Database against Primary Documents** - Once data have been obtained and found to be consistent with the database, we should perform one last check of the database to make sure that it is consistent with the requirements document and the calibration data. This should be carried out under the supervision of the Los Alamos shot engineer.

11. **Check Requirements Document against Physical Measurements Requirements** - One final check should be made between the requirements document and the physical requirements made by external users to see that they are all being met by the system. This is a job for the Los Alamos shot engineer.

12. **Participate in Dry Runs** - Once all the above have been accomplished, the system should be ready for zero time data acquisition. Checklists are provided in the Appendix for preparing for a dry run. Once the system has been verified, it can be run permanently in long term data acquisition mode with arming for each dry run supplied by running the UNIX script, Arm_Systems, on the lachesis machine in the Monitor Room. Remember that this must be done at least 35 minutes prior to the dry run zero time, and preferably a little earlier. The data should be examined for quality, reasonableness of the physical values, duration, and completeness. If there is excessive noise in the data that is not present except during dry runs, the possibility of interaction with

signals from other experiments may have to be investigated. These Monitor Room tasks are to be performed by the Los Alamos shot engineer, but support is expected from the Nevada Engineering support organization.

13. Obtain NTS Coordinates for Sensor Locations and Edit Plotting Database - Once the sensors have been installed, you need to obtain the NTS coordinates (at least north and east coordinates) for each sensor location. This may be done by asking the site engineering group to obtain them, or using the construction station locations to determine them. Once these have been obtained, the plotting database on the lachesis workstation must be edited to include them. This must be done by Los Alamos personnel.

Once these steps have been accomplished, everything is ready for the actual event. Data reduction will take place in the Monitor Room. Quick look plots will be made of all data using the MATLAB script, event_data. These plots will be copied and distributed to customers as quickly and expeditiously as possible. This was completed within 70 minutes for the KISMET test.

Table 1. VEE Codes

Code Name	Function
Archive_Plus_Zero	This is the data collection code. It starts archiving long term data until the user sends a file telling it to prepare for zero time. Then it prepares for zero time data and awaits a TTL trigger into the front panel of the workstation. Once zero time data has been completed and the data sent through the system, it returns to archiving long term data.
Archive_Retrieval	This plots the data in an already existing long term data file. This code exists merely to examine data - high quality plots are provided elsewhere.
Gen_Cal	This code creates or modifies the database of sensor, channel, filename, and amplifier information.
Print_Cal	This prints all database information into two formatted files which may be printed for a permanent record.
VeeDB	This currently experimental code performs the same functions as Gen_Cal, but hopefully in a somewhat more user friendly manner. Furthermore, it can both retrieve and send information to the Pacific amplifiers over the GP-IB port if the user chooses.
Trouble	This updates a log of changes made to the system and allows the user to provide an explanation of the problem that caused the changes.
Make_scan_list	This allows the user to generate the scan lists for any E1413 A/D board in one of several ways.
Multiple_channel_plot	This plots on a continuing strip chart the data for all three components of a triaxial accelerometer array, all the components of a given atmospheric package, and the pore and atmospheric pressure at a given location.
chan_plot	This plots a continuing strip chart of a single channel.

vee_FIFO_data	This collects zero time data according to the current scan list and allows the user to plot channels of this data in arbitrary order.
e1413_cal	This performs a calibration sequence of the selected 64 channel A/D boards on the system.

References

1. MATLAB High Performance Numeric Computation and Visualization Software User's Guide, The MathWorks, Inc., Natick MA, 1992.

2. MATLAB High Performance Numeric Computation and Visualization Software Reference Guide, The MathWorks, Inc., Natick MA, 1992.

3. Robert Helsel, Graphical Programming a Tutorial for HP VEE, Prentice Hall PTR, Upper Saddle River NJ, 1995.

4. Using HP VEE-Engine and HP VEE-Test, 2nd Edition, Hewlett Packard (HP Part No. E2100-90011), 1992.

5. HP VEE-Engine and HO VEE-Test Reference, 2nd Edition, Hewlett Packard (HP Part No. E2100-90013), 1992.

6. Model 9355 Operation and Maintenance Manual, Pacific Instruments, Inc., Concord CA.

7. HP 75000 Series C 64-Channel High Speed Scanning A/D HPE1413A and B User's Manual, 2nd Edition, Hewlett Packard (HP Part No. E1413-90003), 1994.

8. Robert Deupree, LACHESIS Software Report, DX-12N-95-28, 1995.

Appendix: LACHESIS Checklists

A number of checklists have been developed to perform many of the tasks outlined in Section IV. These are included here along with a checklist overview which explains the functions of each of the checklists.

LACHESIS Checklist Overview

The checklists are meant to guide the user through most phases of obtaining zero time data for an event with the LACHESIS. However, it is assumed that the sensors are already in place, that the sensor connections to the individual systems have been checked and corrected if necessary, and that communications and data collection have been routinely exercised by running the systems in archive mode and sending the data to the lachesis workstation in the Monitor Room.

There are several checklists that have been composed. These may be summarized as follows:

Pre Dry Run Checklist - This includes things that should be completed prior to dry runs being begun. These include creating the database on each system, obtaining NTS coordinates from DX-14 (in many applications approximate ones will do), and completing the plotting information database on the lachesis workstation in the Monitor Room. These should be performed by Los Alamos personnel.

Amplifier Checklist - This involves a complete run through of the settings on all amplifiers and requires recording of all settings.

Scan List and Trigger List Checklist - This verifies that the scan lists and trigger lists are made and that they meet all the necessary requirements. It furthermore prevents them from being overwritten.

System Database Checklist - This checks the database for each system and asks the user to verify that it is correct.

General Dry Run Checklist - This checklist is expected to be used on all dry runs until the formal dry runs (FDR), such as the GOYDAR and the final dry run. It sets up and provides the zero time data collection, reduction, and plotting. This checklist expects the user to verify the scan lists, trigger lists, etc.

FDR Checklist - This checklist is to be used for a formal dry run. It is very close to the general dry run checklist except that it assumes that much of the preparatory work is known to be correct. It also is run from the Monitor Room, whereas the general dry run probably works best if run from downhole.

Button Up Checklist - This checklist is to be performed between the GOYDAR and the FDR to make sure that everything is correct. It assumes that you have access to the systems underground.

Event Execution Checklist - This checklist is required for the final stages of event preparation (including arming the systems), data reduction, and quick-look plotting.

Pre Dry Run Checklist

[ ] Determine the types and locations of sensors to be fielded on the event. This should be performed by DX-12 personnel in conjunction with customers (e.g., Containment, Health and Safety, DX-14, Verification).

[ ] For each sensor assign a number to make it unique from all other sensors. The following criteria shall be adhered to:

> **1-20** shall be used to designate permanent facility measurements which will not change from event to event (e.g., sensors in the main drift).
>
> **21-29** shall be used for measurements in and around the <u>Sparta</u> racks.
>
> **31-39** shall be used for measurements in and around the <u>Athens</u> racks.
>
> **41-49** shall be used for measurements in and around the <u>Thebes</u> racks.
>
> **51-69** shall be used for Zero room and near environs measurements.

[ ] Create a channel list associating a channel on a system with a sensor. The channel list shall include all systems and channels, a version number, and the date of the most recent version. This shall be performed and updated by Los Alamos personnel as needed.

[ ] For each sensor obtain from EG&G the calibration information. Write on the cal sheet the sensor location, initial the cal sheet, and store it in the EG&G field notebook. Enter the calibration and amplifier information in the database of the appropriate system by logging onto the system, typing

vee

and bringing up the code Gen_cal located in the directory /users/test. See the LACHESIS software manual for instructions on running vee codes.

[ ] When additions or corrections are made to the database on any system, run the vee code /users/test/Print_cal to make a formatted listing of the database, then type

```
cd /users/test/New_calibrate
rcp print_db_file mycenae:/users/test/System_print_db
rcp print_db_file1 mycenae:/users/test/System_print_db_1
```

where **System** is the system you are modifying the database of (e.g., Athens, etc.). Then you may remotely sign on to the mycenae system by typing

```
remote mycenae
```

and then typing (at the prompt)

```
cd /users/test
printr System_print_db
printr System_print_db1
```

You will be required to answer some questions for each run of printr. In this application, select the landscape option. Obtain the paper from the mycenae printer, put it in the correct system notebook, and fill out the database modification form. As an alternative to remotely signing on to mycenae, you may wish to sign on to mycenae when you get back to CP45 and repeat the three commands above.

[ ] For each sensor location, obtain the NTS coordinates. In many cases this can be done just by using the construction station coordinates. These can be converted into NTS coordinates using the coordinates at the intersection of U1a.01 and U1a.02

(N 823322.88, E 677570.95) = U1a.01 CS+815 = U1a.02 CS+000

These measurements are in feet, and must be converted to meters. Other coordinates for which more accurate determinations are required (such as the working point location) can be obtained from DX-14.

[ ] Update the plotting database in the Monitor Room. Here you must sign on as deupree. Type

```
cd bin
vi station_loc.m
```

and use the vi editor commands to modify the database. When you add a measurement type, you must make three alterations. The first is the four character identifier listed under the variable names. The second is a character string under the variable station_info, and the third is the NTS coordinates under coord_n and coord_e. This is one case in which format is very important. Furthermore, the entries must match for each of these variables (i.e., the seventh element in the names array corresponds to the seventh element in the station_info array and the seventh array member of coord_n and coord_e. This is also one of the few areas in which some knowledge is helpful (MATLAB in this case). However, if you keep the format as it already is, you should be all right. It is expected that this task will be carried out by Los Alamos personnel.

[ ] Create the appropriate directories and links for the event. The event directory should be created under /SCSI_disk4 (and should be /SCSI_disk4/EVENT, where EVENT is the test name) and linked to be /users/test/EVENT. This requires root authority (i.e., call Deupree). Also create appropriate subdirectories under /users/test/EVENT, such as /users/test/EVENT/FDR for the final dry run data, if it is desired to keep this. The plan is to store the event data itself in the directory /users/test/EVENT.

[ ] Make the scan list and trigger list for each board. This is done by running the vee code Make_scan_list.

Athens:	Board 1 [ ]	Board 2 [ ]
Sparta:	Board 1 [ ]	Board 2 [ ]
Thebes:	Board 1 [ ]	Board 2 [ ]
Mycenae:	Board 1 [ ]	Board 2 [ ]
Pylos:	Board 1 [ ]	Board 2 [ ]

AMPLIFIER PROGRAMMING RECORD System ______________ Board: 1 2

Channel	Serial Number	G	G*	F	X	V	AB	A	MON	RC	Z	LOCK	AZ
1													
2													
3													
4													
5													
6													
7													
8													
9													
10													
11													
12													
13													
14													
15													
16													
17													
18													
19													
20													
21													
22													
23													
24													
25													
26													
27													
28													
29													
30													
31													
32													
33													
34													
35													
36													
37													
38													
39													
40													

Date ____________ Time ____________ Signature ______________________________

AMPLIFIER LIGHTS RECORD System ______________ Board: 1 2

Channel	OUT	EXC v	EXC ma	CAL	OVLD	PWR	CHAN#	@ERR
1								
2								
3								
4								
5								
6								
7								
8								
9								
10								
11								
12								
13								
14								
15								
16								
17								
18								
19								
20								
21 22								
23								
24								
25								
26								
27								
28								
29								
30								
31								
32								
33								
34								
35								
36								
37								
38								
39								
40								

Date ___________ Time ____________ Signature ________________________________

Scan List and Trigger List Checklist

[ ] For each system list the scan list. Do this by typing

```
more /users/lanl/scan_list_B1_L1
more /users/lanl/scan_list_B2_L1
```

The first command is for Board 1 - the second command is needed only if there is another board. Enter the scan list below.

[ ] Athens Board 1 :______________________________

[ ] Athens Board 2 :______________________________

[ ] Sparta Board 1 :______________________________

[ ] Sparta Board 2 :______________________________

[ ] Thebes Board 1 : ______________________________

[ ] Thebes Board 2 : ______________________________

[ ] Mycenae Board 1 : ______________________________

[ ] Mycenae Board 2 : ______________________________

[ ] Pylos Board 1 : ______________________________

[ ] Pylos Board 2 : ______________________________

[ ] For each system, check to see that the triggering information is correct for the first board by typing

```
more /users/lanl/gen_FIFO_trig_1
```

There are three lines displayed, each having a number at the end. The first line has the time in seconds between sampling adjacent members of the scan list, the second line has the number of times you go through the scan list (a 0 here means that you go until the FIFO memory is filled), and the third has the time between beginnings of the scan list. If there are two boards on the system, repeat the exercise for the second board by typing

```
more /users/lanl/gen_FIFO_trig_2
```

Record the numbers in the table.

Board 1	Athens	Sparta	Thebes	Mycenae	Pylos
Time between samples					
Number of Triggers					
Time between starting scan lists					
Board 2					
Time between samples					
Number of Triggers					
Time between starting scan lists					

[ ] For each board on each system, calculate to see that the number of entries in the scan list multiplied by the time between samples + 30 microseconds is less than the time between triggers (see page 5-90 in HP E1413 Reference Manual). If not, the scan list **must** be recreated until this is true. Check by using the following table:

Board 1	Athens	Sparta	Thebes	Mycenae	Pylos
a:Time between samples					
b:Number of elements in scan list					
a x b					
a x b + 30 microsec					
Time between triggers					
Board 2					
a:Time between samples					
b:Number of elements in scan list					
a x b					
a x b +30 microsec					
Time between triggers					

[ ] When satisfied that all scan and trigger lists are correct, get someone with lanl privelege to sign on as lanl and type the following:

```
chmod u=r,g=r,o=r scan_list*
chmod u=r,g=r,o=r gen_FIFO_trig*
```

Version 1.0

This will prevent anyone from overwriting the scan and trigger lists. Do this for all systems.

[ ]Athens [ ]Sparta [ ]Thebes [ ]Mycenae [ ]Pylos

[ ] After a dry run dataset has been reduced, open each file and note the starting time and the time step (all the way out to the time between triggers) and compile the results in the accompanying table.

[ ] Sign and Date

Date:______________

Signature:_______________________

Table of Times

File	T start	Delta t		File	T start	Delta t

System Database Checklist

There are a number of locations in which information about the event are kept. These include 1) the original channel list, which defines what each channel of each system contains (this is to be made up by Los Alamos personnel), 2) there is the database on each system which includes all information about specific channels, including the information required to reduce the data, 3) the amplifier checklist, which verifies the settings on the amplifiers, 4) the scan list, which contains the sample information for all channels on a system, 5) the calibration sheets for individual sensors, and 6) the database on the lachesis workstation in the Monitor Room which supplies information for the plots. All of these need to be checked with each other to make sure that they are correct (and consistent).

[ ] For each sensor, check the sensor calibration information from the cal sheets with the physical reference, voltage reference, linear coefficient, serial number, and location in each system database.

[ ] Athens [ ] Sparta [ ] Thebes [ ] Mycenae [ ] Pylos

[ ] For each channel, make sure that the system database agrees with the channel list.

[ ] Athens Board 1	[ ] Athens Board 2
[ ] Sparta Board 1	[ ] Sparta Board 2
[ ] Thebes Board 1	[ ] Thebes Board 2
[ ] Mycenae Board 1	[ ] Mycenae Board 2
[ ] Pylos Board 1	[ ] Pylos Board 2

[ ] Make sure that the scan list agrees with the database for each board on each system. In particular, make sure that all channels are covered.

[ ] Athens Board 1	[ ] Athens Board 2
[ ] Sparta Board 1	[ ] Sparta Board 2
[ ] Thebes Board 1	[ ] Thebes Board 2
[ ] Mycenae Board 1	[ ] Mycenae Board 2
[ ] Pylos Board 1	[ ] Pylos Board 2

[ ] Devise design criteria for each transducer type that specify its excitation voltage, gain, cutoff frequency, and scan rates. Store in event notebook and compare to each system database.

[ ] Compare the results of the amplifier checklist with each system database to make sure that they agree (and are correct).

- [ ] Athens Board 1
- [ ] Sparta Board 1
- [ ] Thebes Board 1
- [ ] Mycenae Board 1
- [ ] Pylos Board 1
- [ ] Athens Board 2
- [ ] Sparta Board 2
- [ ] Thebes Board 2
- [ ] Mycenae Board 2
- [ ] Pylos Board 2

[ ] After a dry run, reduce the data and make the plots. Make sure all relevant channels on the channel list produce data files and that there are no extras (unless you want them).

[ ] Check the data from a dry run with the last archive data point prior to the run to check the data for suitability and consistency.

[ ] Take the data plots from a dry run and see that the distances from WP are correct and that the header and axis information is present and correct. If not, you may have to add information to the plotting database in the matlab file station_loc.m

[ ] Store some dry run data in the event notebook and in subdirectories of the event to provide data for noise characterization and potentially other comparisons.

[ ] Sign and date.

Date: ________________________

Signature: ________________________

General Dry Run Checklist

[ ] Verify that the scan and trigger lists exist and can be read by typing (the spaces for UNIX commands are important)

```
ll /users/lanl/scan_list*
ll /users/lanl/gen_FIFO_trig*
```

The return messages should have an r in the second, fifth, and eighth places. Thus, the beginning should read

-r**q**-r**q**-r**q**-

where **q** may be either w or -. There should be two scan list files (scan_list_B1_L1 and scan_list_B2_L1) and two trigger lists (gen_FIFO_trig_1 and gen_FIFO_trig_2) if two E1413 boards are being used.

[ ] Athens [ ] Sparta [ ] Thebes [ ] Mycenae [ ] Pylos

[ ] Verify that the file Archive_to_Zero exists in the proper directory and has read and write permission for everyone by typing

```
ll /users/lanl/bin/Archive_to_Zero
```

and seeing that the return begins -rw-rw-rw

[ ] Athens [ ] Sparta [ ] Thebes [ ] Mycenae [ ] Pylos

[ ] Verify that the file Archive_to_Zero has a 0 in it by typing

```
more /users/lanl/bin/Archive_to_Zero
```

and seeing that a 0 is returned.

[ ] Athens [ ] Sparta [ ] Thebes [ ] Mycenae [ ] Pylos

[ ] Check to see that the three options on the archiver are properly set. The first should be to send data to lachesis, the second to reduce data, and the third to one board or two, depending on how many E1413 boards are on the system. If all of these are not correctly set, or if the archiver is not running, bring up vee, and then bring in Archive_Plus which resides in /users/test, set the three parameters correctly, and begin the archiver.

[ ] Athens [ ] Sparta [ ] Thebes [ ] Mycenae [ ] Pylos

[ ] Check to see that the archiver is collecting data from both boards (if there are two boards on the system). This may be done by clicking a number in the panel on the lower right and entering a number between 100 and 163 for the first board and 200 and 263 for the second board. Check the responses in the data and comment boxes to see that they change and make sense.

[ ] Athens [ ] Sparta [ ] Thebes [ ] Mycenae [ ] Pylos

[ ] From any system, log on remotely to the lachesis workstation at least forty minutes before you wish to trigger the zero time data collection. This is done by getting into an unused window and typing

remote lachesis

and following the instructions. When you have finished the logging on and are ready to arm all systems, type

/users/deupree/bin/Arm_systems

and see if the responses match what the text indicates that they should. If they do not, keep the screen visible and notify Deupree of the problem. Check to see that all systems that are supposed to be taking zero time data have been listed. Exit from lachesis if logged on remotely.

[ ] When each system finishes collecting the 100 data points that make up each archiving average and the panel at the lower right of the archiver disappears, see that the Access light on the front panel of the V382 slot zero controller blinks about once per second (it may take a few seconds for this to happen). If it does, then the system is awaiting a trigger to collect the zero time data. If not, something is wrong and professional help (i.e., Deupree) is required to figure it out. If you are expecting a trigger from the FIDU boxes, **do not** play with the coaxial cable (particularly unplug and replug it) as this may trigger the system.

[ ] Athens [ ] Sparta [ ] Thebes [ ] Mycenae [ ] Pylos

[ ] The systems are now ready to be triggered. This is done by sending a ttl signal into the front panel of the V382 controller in the bnc connection marked Trigger In. When the signal is sent, the Access light should continue

blinking about once a second for a few seconds, and then the panel at the lower right of the archiver should reappear. This means that the archiver has returned to archiving mode.

[ ] Athens [ ] Sparta [ ] Thebes [ ] Mycenae [ ] Pylos

[ ] Log on remotely to the lachesis workstation (see previous step) and type

```
ls /users/test/EVENT_DATA/*
```

This will list the files for each subdirectory (given by the system names). For each system there should be two files per board one begins with Zero_Time_Data and the other with Reduce_Zero_Data). If they are not all present on the lachesis workstation, check to see if they are present on the appropriate system.

[ ] Athens [ ] Sparta [ ] Thebes [ ] Mycenae [ ] Pylos

[ ] Return to the Monitor Room and reduce the data. Log on as test and move to one of the subdirectories where the uploaded data is stored by typing

cd /users/test/EVENT_DATA/**System**

where **System** is the name of one of the systems on the test (e.g., Athens). Then run, once for each board on the system, the data reduction code by typing

```
Reduce_Zero_Time_Data
```

and following the instructions. The data reduction takes about three minutes per board if all the FIFO memory is filled.

Athens:	Board 1 [ ]	Board 2 [ ]
Sparta:	Board 1 [ ]	Board 2 [ ]
Thebes:	Board 1 [ ]	Board 2 [ ]
Mycenae:	Board 1 [ ]	Board 2 [ ]
Pylos:	Board 1 [ ]	Board 2 [ ]

[ ] When you have completed all boards on a system, change the names of the uploaded files by typing

mv Zero_Time_Data_1 **System**_Zero_Time_Data_1
mv Reduce_Zero_Data_1 **System**_Reduce_Zero_Data_1

If there are two boards on the system, type

```
mv Zero_Time_Data_2 System_Zero_Time_Data_2
mv Reduce_Zero_Data_2 System_Reduce_Zero_Data_2
```

where **System** is the name of the system that corresponds to the subdirectory you are in.

Athens:	Board 1 [ ]	Board 2 [ ]
Sparta:	Board 1 [ ]	Board 2 [ ]
Thebes:	Board 1 [ ]	Board 2 [ ]
Mycenae:	Board 1 [ ]	Board 2 [ ]
Pylos:	Board 1 [ ]	Board 2 [ ]

[ ] Then copy all the files to the appropriate directory for plotting by typing

```
cp /users/test/EVENT_DATA/System/* /users/test/EVENT
```

where **EVENT** is the name of the test (e.g., KISMET). Repeat this step for all systems on the test.

Athens:	Board 1 [ ]	Board 2 [ ]
Sparta:	Board 1 [ ]	Board 2 [ ]
Thebes:	Board 1 [ ]	Board 2 [ ]
Mycenae:	Board 1 [ ]	Board 2 [ ]
Pylos:	Board 1 [ ]	Board 2 [ ]

[ ] Move to the appropriate directory and bring up the MATLAB plotting package by typing

```
cd /users/test/EVENT
matlab
```

where **EVENT** is the name of the test. Wait for the MATLAB prompt (>>) and then type

```
event_data
```

to plot all zero time data. Follow the instructions. Be sure you pick to print all plots on the Laserjet 4P printer (pick laserjet3 option when asked). This printer is the system printer. You do not wish to print the plots on the Laserjet 2P printer because it will take a <u>long</u> time.

[ ] Once the plots have come out, you need to check them to see that they are correct. Particularly, you need to

check 1) are all the sensors represented? (if not, then check to see if the file is in /users/test/**EVENT** - if not, it is probably not included in the scan list, if so, there is probably something wrong with the plotting database), 2) are the distances from the WP correct? (if not, there is something amiss in the plotting database), 3) are the data reasonable when compared to the archive data?

[ ] When you are done, you should clean up the files by typing the following

```
rm /users/test/*_check
rm /users/test/EVENT_DATA/System/*
```

where **System** is any system on the test (proceed through all the systems). If you wish to save the data from this dry run permanently, type the following

```
ls -d /users/test/EVENT
```

which will indicate the directories that are available to you. Pick the appropriate one and type

```
mv /users/test/EVENT/* /users/test/EVENT/subdir
```

where subdir is the appropriate subdirectory. If there are no appropriate subdirectories, someone with root authority must make them for you.

[ ] To get the data from the archiver at the time of the test, type the following

```
cd
matlab
```

and at the matlab prompt, type

```
find_pre_zero
```

and follow the instructions. Once the matlab prompt has returned, type exit to leave matlab and then type

```
printr pre_zero_values.sum
```

Check this against the plots for all sensors to see that they agree.

Formal Dry Run Checklist

It is assumed by this time that there have been several successful dry runs accomplished and that the LACHESIS is generally functioning properly. This checklist carries the user through a formal dry run (e.g., GOYDAR, FDR) when the LACHESIS is to be controlled from the Monitor Room. However, people can be downhole to observe the process. It is expected that the vee program Archive_Plus_Zero is up and functioning prior to the beginning of the dry run.

[ ] Enter the name of the run (GOYDAR, etc.) , the date, and your name.

Run Name:______________________________
Date and Time:______________________________
Name:______________________________

[ ] Log on to the lachesis workstation in the Monitor Room as the user test. As you log on check to see (via the window that comes up) that you are collecting data on all systems that are participating on the event.

[ ] Remove any previous dry run files and other checking information by typing

rm *_check
rm /users/test/EVENT_DATA/**SYSTEM**/*

where **SYSTEM** refers to any system (e.g., Athens) on the event. Repeat the second command for all systems on the test.

[ ] Approximately 45 minutes prior to zero time, arm the systems on the event by typing

Arm_systems

and checking the results as requested. If the results are as indicated, then the systems are armed.

[ ] If anyone is observing the dry run from downhole, verify with them that the system goes into zero time data collection mode at the end of a 100 point archive data collection set.

[ ] About one minute after zero time has passed, see if the data and data reduction files have been successfully

uploaded to the lachesis workstation by typing

```
ls /users/test/EVENT_DATA/*
```

There should be two files for each E1413 board on each system - one beginning Zero_Time_Data (which contains the raw data off the E1413 board) and another beginning Reduce_Zero_Data (which contains the information required to produce the final data). Check to see that all these files are present.

Athens:	Board 1 [ ]	Board 2 [ ]
Sparta:	Board 1 [ ]	Board 2 [ ]
Thebes:	Board 1 [ ]	Board 2 [ ]
Mycenae:	Board 1 [ ]	Board 2 [ ]
Pylos:	Board 1 [ ]	Board 2 [ ]

[ ] Move to the subdirectory for one of the active systems by typing

cd /users/test/EVENT_DATA/**SYSTEM**

where **SYSTEM** is any of the systems (e.g., Athens). Then, for each board on that system, run the C code to reduce the data by typing

```
Reduce_Zero_Time_Data
```

and follow the instructions. This can take about three minutes per board if the entire FIFO is filled. You may do different systems simultaneously by using different windows.

Athens:	Board 1 [ ]	Board 2 [ ]
Sparta:	Board 1 [ ]	Board 2 [ ]
Thebes:	Board 1 [ ]	Board 2 [ ]
Myceane:	Board 1 [ ]	Board 2 [ ]
Pylos:	Board 1 [ ]	Board 2 [ ]

[ ] In order to collect all the files in one place, you need to change some file names. In each subdirectory type

mv Zero_Time_Data_1 **SYSTEM**_Zero_Time_Data_1
mv Reduce_Zero_Data_1 **SYSTEM**_Reduce_Zero_Data_1

where **SYSTEM** is the subdirectory you are in. If there is more than one board on this system, type

mv Zero_Time_Data_2 **SYSTEM**_Zero_Time_Data_2
mv Reduce_Zero_Data_2 **SYSTEM**_Reduce_Zero_Data_2

Athens:	Board 1 [ ]	Board 2 [ ]
Sparta:	Board 1 [ ]	Board 2 [ ]
Thebes:	Board 1 [ ]	Board 2 [ ]
Myceane:	Board 1 [ ]	Board 2 [ ]
Pylos:	Board 1 [ ]	Board 2 [ ]

[ ] For each system on the event, move the data to a common directory by typing

cd /users/test/EVENT_DATA/**SYSTEM**
cp * /users/test/**EVENT**

where **EVENT** is the name of the event.

Athens [ ] Sparta [ ] Thebes [ ] Pylos [ ] Mycenae [ ]

[ ] Move to the event directory and enter matlab by typing

cd /users/test/**EVENT**
matlab

[ ] When the matlab prompt appears (>>), type

event_data

and follow the instructions. In particular, make sure that you pick the Laserjet 4 printer for the plots.

[ ] When the plots are done, exit matlab (by typing exit) and move back to the /users/test directory, and reinvoke matlab by typing

cd
matlab

[ ] Run the script to find the last archive set of values before the dry run for each channel by typing

find_pre_zero

[ ] When the matlab prompt returns, exit matlab and print the results by typing

printr pre_zero_values.sum

and selecting portrait and the Laserjet 4 when queried.

[ ] Examine the data for acceptability and put all the plots and the printed file from the previous step in the event notebook along with this checklist.

[ ] Make sure that the archiver has begun taking archive data again. Do this by typing

```
ls
```

and picking a data set from each system on the event. Then type

tail **DATASET**

where **DATASET** is the name of the archive data set (since the names are long, you will probably want to use the mouse to latch onto the name). See that the time of the most recent entry is after the event (this will probably take about 35 minutes for the most heavily used systems).

Button Up Checklist
(Assumes Underground Access)

[ ] On the FIDU boxes, pull the "clear latch lights" switch to make sure that the FIDU received light is off.

Athens [ ] Sparta [ ] Thebes [ ] Mycenae [ ] Pylos [ ]

[ ] Verify that the system is logged on as the user test and that the current directory is /users/test. This may be done by creating a window, moving the mouse pointer into it, clicking the left mouse button and typing

pwd

If any reply other than /users/test appears, you are logged on as a different user. In this case, stop all executing programs, log off the system, and log back on as test.

[ ]Athens [ ]Sparta [ ]Thebes [ ]Mycenae [ ]Pylos

[ ] Check to see that the scan lists are correct. For the first A/D board type

more /users/lanl/scan_list_B1_L1

The first line contains the number of entries in the scan list, while the second contains the scan list itself. If the system has two boards, verify that the scan list for the second board is correct by typing

more /users/lanl/scan_list_B2_L1

(Figure out a mechanism for writing the scan list down.)

Athens:	Board 1 [ ]	Board 2 [ ]
Sparta:	Board 1 [ ]	Board 2 [ ]
Thebes:	Board 1 [ ]	Board 2 [ ]
Mycenae:	Board 1 [ ]	Board 2 [ ]
Pylos:	Board 1 [ ]	Board 2 [ ]

[ ] Check to see that the triggering information is correct for the first board by typing

more /users/lanl/gen_FIFO_trig_1

There are three lines, each having a number at the end.

The first line has the time in seconds between sampling adjacent members of the scan list, the second has the number of times you go through the scan list (a 0 here means that you go until the FIFO memory is filled), and the third has the time between beginnings of the scan list. If there are two boards on the system, check the second by typing

more /users/lanl/gen_FIFO_trig_2

Record the numbers in the table.

Board 1	Athens	Sparta	Thebes	Mycenae	Pylos
Time between samples					
Triggers					
Time between starting scan list					
Board 2					
Time between samples					
Triggers					
Time between starting scan list					

[ ] Obtain a final copy of the database by moving to an open window, typing

vee

and bringing up the vee code Print_cal when vee appears. This last is accomplished by first clicking on "File" in the upper left hand corner and then on "Open ..." when the submenu appears. This will produce a new window in which you can enter Print_cal. When the vee code appears, click on the Start button in the panel and wait for the

Version 1.2

number 127 and the word Completed to appear. Once both have occurred, exit vee (by clicking on "File", then "Exit", and finally on "No, clear" in the new window) and type

```
rcp New_calibrate/print_db_file lachesis:/users/test/System_db
rcp New_calibrate/print_db_file1 lachesis:/users/test/System_db_1
```

where **System** is the name of the system you are logged onto (e.g., Athens). All of this may be performed even if the archiver is running.

[ ]Athens [ ]Sparta [ ]Thebes [ ]Mycenae [ ]Pylos

[ ] Verify that the file Archive_to_Zero is in /users/lanl/bin and had rw permission for everyone by typing

ll /users/lanl/bin/Archive_to_Zero

and seeing that the return begins with

-rw-rw-rw

[ ]Athens [ ]Sparta [ ]Thebes [ ]Mycenae [ ]Pylos

[ ] Verify that the file Archive_to_Zero has a 0 in it. Do this by typing

more /users/lanl/bin/Archive_to_Zero

and seeing that a 0 is returned.

[ ]Athens [ ]Sparta [ ]Thebes [ ]Mycenae [ ]Pylos

[ ] Verify that the system is on the network by clicking on an open window and typing

ping lachesis

and then <Crtl>C to exit after some pings have taken place.

[ ]Athens [ ]Sparta [ ]Thebes [ ]Mycenae [ ]Pylos

[ ] If the archiver is not already running, get into vee and bring up the vee code Archive_Plus_Zero (see steps for bringing up Print_cal).

[ ]Athens [ ]Sparta [ ]Thebes [ ]Mycenae [ ]Pylos

[ ] Set or verify that the three options appearing on the panel are correctly set. The first two should be to send data to lachesis and Reduced data. The number of boards refers to the number of E1413 64 channel scanner boards on the system. Once the three selections are correctly made, start the archiver by pressing the Run button at the upper right.

[ ]Athens [ ]Sparta [ ]Thebes [ ]Mycenae [ ]Pylos

[ ] Verify that the archiver is being run from /users/test by checking the current directory in the lower left part of the panel. If the current directory is not /users/test, exit vee, log off the system, and log back on as the user test. If the words current directory do not appear at the bottom left corner of the panel, you are not using the correct archiving program and need to repeat the previous two steps.

[ ]Athens [ ]Sparta [ ]Thebes [ ]Mycenae [ ]Pylos

[ ] Verify that the archiving program is collecting data by seeing that the numbers change in the Channel Data display in the panel at the lower right. If the system is collecting data from two boards, examine a channel from each board. This is done by entering a number between 100 and 163 for the first board and between 200 and 263 for the second board in the Desired Channel Number display. A number is entered in this display by clicking the left mouse button when the pointer is over the number in the display, entering the desired number, and hitting the carriage return. NOTE that failing to hit the carriage return will lock up data collection.

[ ]Athens [ ]Sparta [ ]Thebes [ ]Mycenae [ ]Pylos

[ ] Verify that the counter on the main panel is relatively small (<2000). If not, repeat the previous three steps.

[ ]Athens [ ]Sparta [ ]Thebes [ ]Mycenae [ ]Pylos

[ ] Verify that the rack temperatures are within acceptable limits. Do this by using the previous step with the appropriate channel numbers for the rack temperature measurements. If you do not know these, consult the database print out. Enter the results in the table below.

System	Amp 1 Rack	Amp 2 Rack	HP Rack
Athens			
Sparta			
Thebes			
Mycenae			
Pylos			

[ ] Move the mouse pointer to any open space in the main panel and click the left mouse button.

[ ]Athens [ ]Sparta [ ]Thebes [ ]Mycenae [ ]Pylos

[ ] Verify that the flow meter in the 48" pipe is pointing in the correct direction (or has been removed and the hole plugged).

[ ] Verify that all power cords are firmly connected. If the chairs are still in the alcoves, lay them down. Close all rack doors.

Event Execution Checklist

[ ] Log onto the lachesis workstation as the user test.

[ ] Verify that all systems are sending data by examining the window that appears on the screen during the sign on process. The statement that a system is sending data means that data has been sent to lachesis at least once since midnight.

[ ]Athens [ ]Sparta [ ]Thebes [ ]Mycenae [ ]Pylos

[ ] Verify that each system is currently sending data. Do this by typing

ls *date

where date is the six character designation for the current data (for example, January 13, 1995 would be 011395). Pick one file from the list generated for each system and type

tail filename

where file name is the name of the file. Verify that the last number in the first column is within the last hour (24 hour clock). If this is not the case, you are no longer receiving data from that system. If you are not receiving data, move to an open window and type

ping **System**

where **System** is the name of the system for which you are not receiving data. If the ping is unsuccessful, there is a network problem. If the ping is successful, the system may no longer be collecting data.

[ ]Athens [ ]Sparta [ ]Thebes [ ]Mycenae [ ]Pylos

[ ] Verify that the UNIX script Arm_systems exists and has universal execute privilege by typing

ll /users/deupree/bin/Arm_systems

The first part of the response should be -rwxr-xr-x

[ ] Verify that the file One exists and that the user test

has rw access by typing

ll One

and seeing that the return begins -rw

[ ] Remove extraneous files from previous dry runs by typing

rm *_check

[ ] Remove data from previous dry runs by typing

rm /users/test/EVENT_DATA/**System**/*

where **System** is any system on the test (e.g., Athens).

[ ]Athens [ ]Sparta [ ]Thebes [ ]Mycenae [ ]Pylos

[ ] Approximately 45 minutes (but definitely more than 30 minutes) before zero time, arm all systems to take zero data by typing

Arm_systems

Check the results as indicated.

[ ] In the event of a long (user discretion is required to define) delay, the systems may be returned to archiving mode by typing

Disarm_systems

Depending on the amount of time elapsed, one may wish to rearm the systems by executing the previous step or to begin this checklist anew.

[ ] Once the test has been executed and about one minute has passed, see if the data and data reduction files have been successfully uploaded to lachesis by typing

ls /users/test/EVENT_DATA/*

There should be two files for each E1413 board on each system: Zero_Time_Data_n (which contains the voltages read off the E1413 board) and Reduce_Zero_Data_n (which contains the data reduction information), where n refers to the board number (e.g., 1 or 2).

Athens: Board 1 [ ] Board 2 [ ]

Sparta:	Board 1 [ ]	Board 2 [ ]
Thebes:	Board 1 [ ]	Board 2 [ ]
Mycenae:	Board 1 [ ]	Board 2 [ ]
Pylos:	Board 1 [ ]	Board 2 [ ]

[ ] Move to the subdirectory for one of the systems by typing

cd /users/test/EVENT_DATA/**System**

where **System** is one of the system names. Then run (once for each board on that system) the C code to reduce the data by typing

Reduce_Zero_Time_Data

and following the directions. This can take about three minutes per board. When this is completed, change the file names by typing

mv Zero_Time_Data_n **System**_Zero_Time_Data_n
mv Reduce_Zero_Data_n **System**_Reduce_Zero_Data_n

where **System** refers to the system whose subdirectory you are in and n is the board number. Then copy the files to the event directory by typing

cp * /users/test/**EVENT**

where **EVENT** is the name of the test. Repeat for all boards on the system. To move to another system, repeat all the instructions in this step.

Athens:	Board 1 [ ]	Board 2 [ ]
Sparta:	Board 1 [ ]	Board 2 [ ]
Thebes:	Board 1 [ ]	Board 2 [ ]
Mycenae:	Board 1 [ ]	Board 2 [ ]
Pylos:	Board 1 [ ]	Board 2 [ ]

[ ] Enter the MATLAB plotting package by typing

cd /users/test/**EVENT**
matlab

where **EVENT** is the name of the test.

[ ] Plot all zero time data by typing

event_data

and following the instructions. In particular, make sure you pick to print all plots on the Laserjet 4P printer. Do **not** print the plots on the Laserjet 2P because it will take a very long time.

[ ] Move back to the /users/test directory by typing

cd

[ ] Enter the MATLAB plotting package by typing

matlab

[ ] Run the script to find the last archive set of values for each channel taken before zero time by typing

find_pre_zero

and entering the day and time of test execution when queried. When this has been completed (matlab returns with the prompt >>), exit matlab by typing

exit

[ ] Print the results of this by typing

printr pre_zero_values.sum

and select portrait and the Laserjet 4 when queried.

[ ] A copy of the data plot should be made available to the test director. A copy of the data and the printed file pre_zero_values.sum should be given to the Los Alamos DX-12 person in charge of final data reduction.

[ ] You may be asked to provide real time (faster than the 30 minute archiving average) information for pressurizing the complex or for reentry information. If this is the case, make sure that you stop the archiving from downhole and restart it remotely on each system from the lachesis workstation in the Monitor Room. This is done by typing

Remote_Halt

and following the instructions. This will allow you to selectively remotely halt systems.

[ ] For each system, check the last entry in the file archive_broken to see if the archiver has actually stopped (this requires up to 35 minutes on those systems with two E1413 boards and up to 20 minutes on those

systems with one E1413 board). This is done by remotely logging onto the system and typing

```
tail /users/test/archive_broken
```

and reading the time of the last entry. Repeat this approximately every five minutes until the last time entry is near the current time. This will mean that the archiver has finished processing and has stopped. When the time is near current, you may bring up the archiver remotely in the Monitor Room for that system. This will allow you to use the current display window in the Monitor Room to provide interested parties with real time data.

[ ] Athens [ ] Sparta [ ] Thebes [ ] Pylos [ ] Mycenae