2002 NASA/ASEE SUMMER FACULTY FELLOWSHIP PROGRAM

JOHN F. KENNEDY SPACE CENTER
UNIVERSITY OF CENTRAL FLORIDA

THE ELECTRONIC NOSE TRAINING AUTOMATION DEVELOPMENT

Nathan Schattke
Research Associate
BCPS Department
Illinois Institute of Technology
Rebecca Young

ABSTRACT

The electronic nose is a method of using several sensors in conjunction to identify an unknown gas. Statistical analysis has shown that a large number of training exposures need to be performed in order to get a model that can be depended on. The number of training exposures needed is on the order of 1000.

Data acquisition from the noses are generally automatic and built in. The gas generation equipment consists of a Miller-Nelson (MN) flow/temperature/humidity controller and a Kin-Tek (KT) trace gas generator. This equipment has been controlled in the past by an old data acquisition and control system. The new system will use new control boards and an easy graphical user interface. The programming for this is in the LabVIEW G programming language. A language easy for the user to make modifications to.

This paper details some of the issues in selecting the components and programming the connections. It is not a primer on LabVIEW programming, a separate CD is being delivered with website files to teach that.

THE ELECTRONIC NOSE TRAINING AUTOMATION DEVELOPMENT

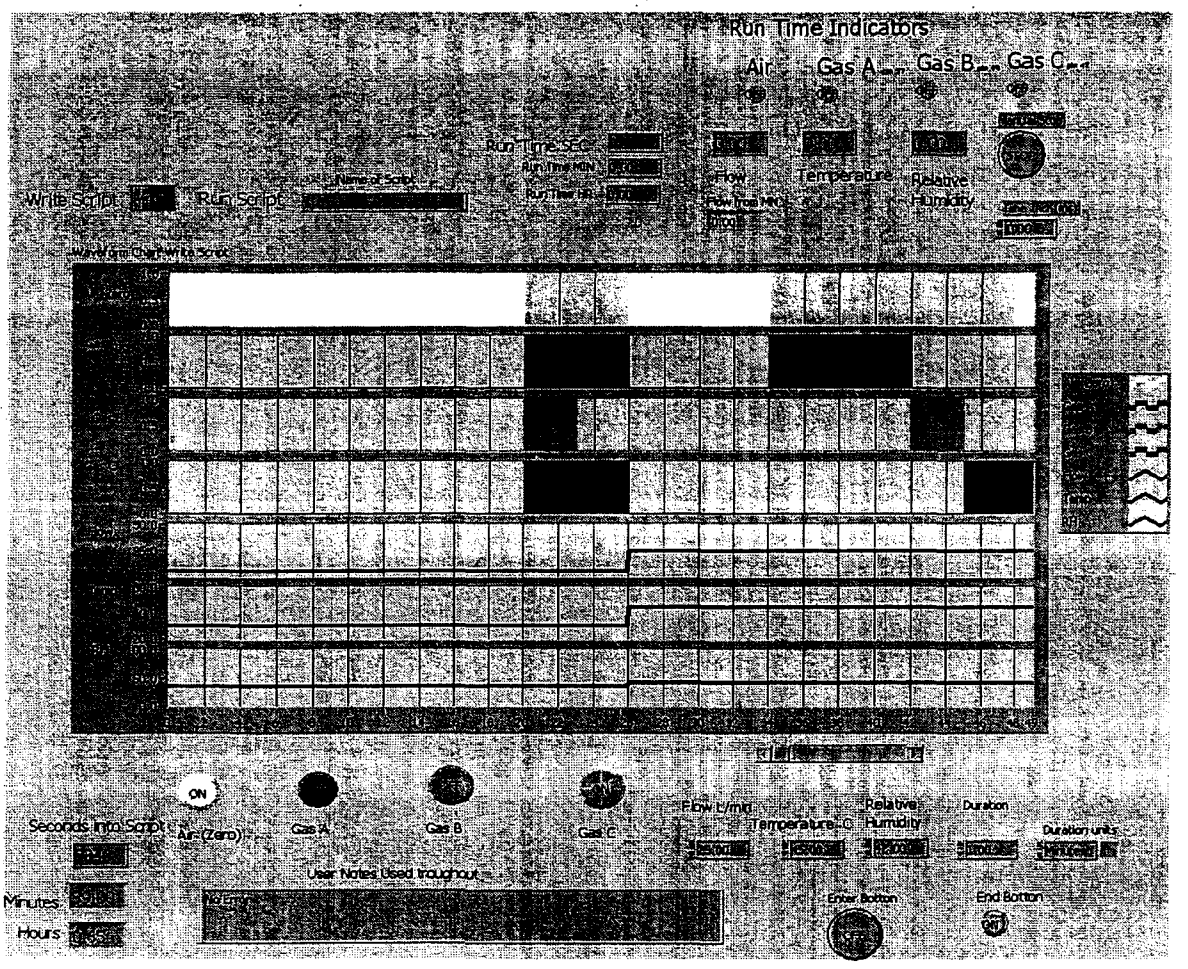
Nathan Schattke

Introduction

Training the electronic nose is a time consuming task. The statistically significant number of training exposures can run into the thousands. In order to get that many tests computer control of the gas generation equipment is required. The Chemical Instrumentation and Processes Lab in the Operations & Controls building at the Kennedy Space Center is an advanced laboratory for evaluating chemical sensors, especially those sensors relating to spaceport related gasses. The most important gas is Hydrazine, an excellent propellant but with a threshold limit value of 10 parts per billion in air. The equipment in the lab for creating a stream of this gas, and others, are several Kin-Tek Span Pac 361 automated gas standards generators. These produce a trace gas by mixing a minute flow of component vapor into a much larger flow of dilution gas.[1] Molecular permeation of vapor through a polymeric membrane is the controlling mechanism. That is, they use a small section of a porous plastic between a very high concentration sample and a gas stream. In order for this to be reproducible the permeation source must have a steady flow of gas across it and a steady temperature. To use the gas one needs to plumb a sample line from the outlet of the unit towards the sensor to be tested. The disadvantage of this is that the gas only comes from three sources, each of a specific concentration. Another piece of equipment in the lab helps to dilute those three concentrations down to a much wider concentration range. This second instrument is the Miller-Nelson Research inc. Flow-Temperature-Humidity control system HCS-301. This has analog control of pure air flow from 5 to 50 liters per minute, Temperature range of 20 to 55 C, and Relative Humidity of 5% or 20 to 90%. A precision mass flow controller, a heated water chamber and a dry heater achieve these parameters.

Training the electronic nose is a time consuming process. It must be trained with many different parameters including all the different gasses at different temperatures, concentrations, and humidity conditions. Previously the control for parts of this was performed with a combination of 15-year-old data acquisition and control and manual operation. This paper describes an implementation of National Instruments' LabVIEW to control these two instruments. Also, future improvements to the system are explained.

[1] Kin-Tek labs Operating Instructions



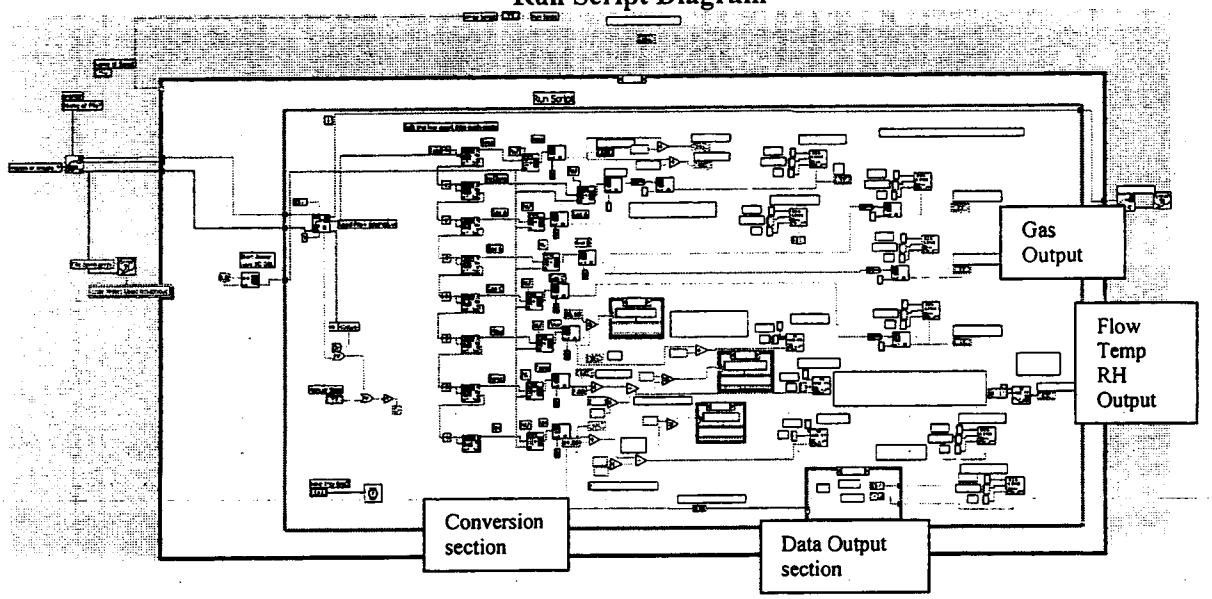
Use

The program is called "Read and Write.vi" in the "KinTek and MN Control folder", it must also have the library file in the same directory. The program is made to build scripts then run them. Choose the "Operate Value" tool, it looks like a hand with the index finger pointing up, from the tools palette. Use this to operate the slide switch in the top left corner of the front panel. The default position is write script, click on the right side to run a script. Start the program by pressing the run button in the lower left of the menu bar on top. The first thing you will be asked is the name and location of the file.

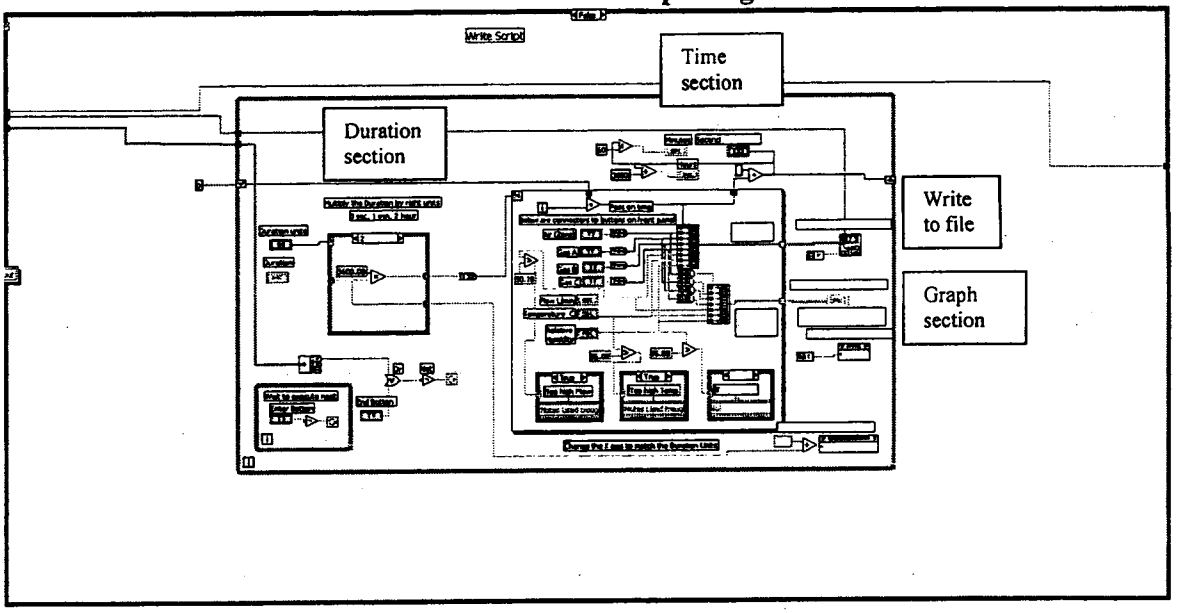
When writing the script, choose the settings for the next step. That is: what gasses will be used, what flow rate, temperature, relative humidity, duration, and duration units. If this is the last step in the script press the End button. Otherwise press the Enter button and continue to make changes after each step. When you are ready for the last step press the End button, then the Enter button twice. The file created is a text tab delimited file with the following columns: Time (starting at 0 sec), Zero/Air (0 or 1), Gas A, Gas B, Gas C, Flow (0-50 L/min), Temperature (0-55C), Relative Humidity (20-90%). There is no header or anything else in the file.

After the program stops, you can change to Run Script. In this case when you run the program with the arrow the computer will ask for a file to run. The progress of the run is shown in the text boxes in the upper right hand corner. I was unable to create a graph that would show this. Time is to the left in Seconds, Minutes and Hours. The LED type indicators light when the particular gas is on, the MN values are below those. A manual stop button is ready to end the program smoothly if a run needs to be aborted prematurely. The time increment button is set in milliseconds. It will almost never be touched, however, if you wished to use shorter files and run each step as minutes or hours you would use 60,000 or 3,600,000 respectively.

Run Script Diagram



Write Script Diagram



Logic Flow

The program is set up to do a few things then choose which condition to do as this run, either Read or Write. Within each of these cases the program operates on a one second per step basis, either reading or writing the values for the parameters.

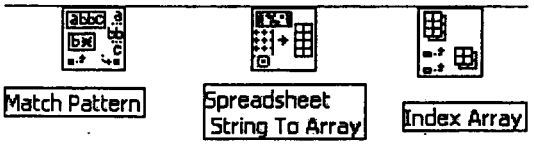
The first logical split is whether this run of the program will be Read or Write. The program opens on old tab delimitated text file (Read) or creates a new file (Write).

For Read, also called Run Script, the file is read one line at a time using a While loop, each

iteration of the loop takes 1 second (controlled by the little watch in the lower left corner in the diagram).

A While loop.

The line is translated as follows. It is read as a script, this is split into before and after the tab symbol eight times, anything left after that in a line is ignored. Those sections of the script are each converted into double byte real number 1 dimensional arrays. The first (#0) byte of that is stripped off and translated into the appropriate type of number (Boolean, integer, or real). This number is finally presented to the DAQ system.



These blocks are used for this translation.

The data acquisition and control (DAQ) system control is performed in the program by the black and white square block with either "Dig Line" or "AO One Pt" written inside.

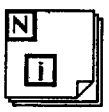
Digital out

Analog out

This is the heart of the program and deserves a bit of explanation. Besides the data feeding it there are three parameters: Device, Channel, and Line. Device can be found by running either NI-DAQ or "Measurement & Automation " programs in the National Instruments directory. When that program is run, the current DAQ boards in the computer are found and identified (you can even test them from this program). For all the outputs currently device 2 is used which is a National Instruments Co. NI 6711 (pn 777740-01). This is an Analog Output – 1 MS/S/Channel, 12-Bit, 4 Analog output channels. There are also 8 bits of digital data that can be either input or output. A line is the value of a singe bit of that word. We will use the analog controls to change the flow, temperature and relative humidity. We will use the digital controls to turn the gasses on and off and change all the relays.

A few more sections of the Read need to be explained. The relays for Zero/Air are controlled near the top. Relay A is always on; relay B is turned on when air is on, otherwise it is off. Both relays off control Standby but that is it is not desired. The relay control for the Wet/Split/Dry control is in the bottom right of the run script while loop. This is a conditional control with the relays set depending on the requested RH. The hardware is not ready for this yet. The relays inside the humidifier are 24 volts actuated and the highest these types of boards give is 10 volts.

The Writer Script section is easier. This section builds the file using iterations of a While loop with a nested For loop.



A For loop.

A Conditional structure

The choice of what state to have the buttons at, is done by the front panel controls. I'll explain about the front panel in the next paragraph. After the state of each part is chosen the Enter button is pressed on the front panel and the while loop iterates once. The duration units control a Conditional structure that multiplies the Duration by 1, 60 or 3600 for seconds, minutes, or hours respectively. This feeds the number of times to do the For loop to iterate and converts everything down to seconds. Within the For loop the blue line (integer) on top brings in the last time written and adds the current iteration number of

the For loop (0 to start). This value is the first of several numbers put into an array. For the Air and Gas buttons note that the boolean (green) values have been converted to integer (blue) and further down another part of that line has been converted to real (orange). This data is grouped together into two arrays. The top array is for writing the data file; the bottom is for the graph. The graph needs to be fed all real numbers but the writing file can take reals or integers but not booleans. The one-dimensional array forms a two-dimensional array at the border of the For loop by "indexing" (right click on the little box at the edge to activate). Outside the For loop the two-dimensional array is fed to a sub program called "write to spreadsheet file.vi". This converts all the values into their text equivalents with tabs separating all the numbers and an end of line symbol at the end. Then the while loop waits until the Enter button is pressed before doing it all again, giving you time to change any parameters on the front panel. The while loop ends when the End button is pressed. This part of the program is not performing as I would like and the while loop executes one more time because the Enter button is then pressed and then the false state ending command is given to the while loop end condition.

The front panel is the user interface. The "Diagram" is the program. For each element on the front panel a block is put in the diagram and wires of data are fed to it or come from it. The connection between the two can be found by: using the arrow tool, selecting the front or diagram part, right clicking and choosing "find terminal". The properties of the chart are controlled from the front panel, unless a "property node" is created in the diagram. These property nodes allow you to control or read parts of the graph from the diagram. In this program the X-axis scale is set with the duration units.

Hardware connections. The DAQ boards are plugged into the computer slots. Cables go from there to a connector block. The wires from the cables connect to certain pins on this connector block as defined in the drawing below. Device 1 is the 6023, device 2 is the 6711. The other end of the cable is plugged into the appropriate machine.

Opportunities for improvement

The graph for running the script was attempted. I tried to put a waveform chart within the while loop. This caused a fault in the program that shut the program down with a note to call National Instruments.

There must by a way to select portions of the write graph then copy and paste them. I was not able to find it. Also in the write section the ending switch needs to be more immediate, the file is written to one more time, this could cause problems. Perhaps one fix is to force the feed to N of the For loop to zero when the end button is pressed.

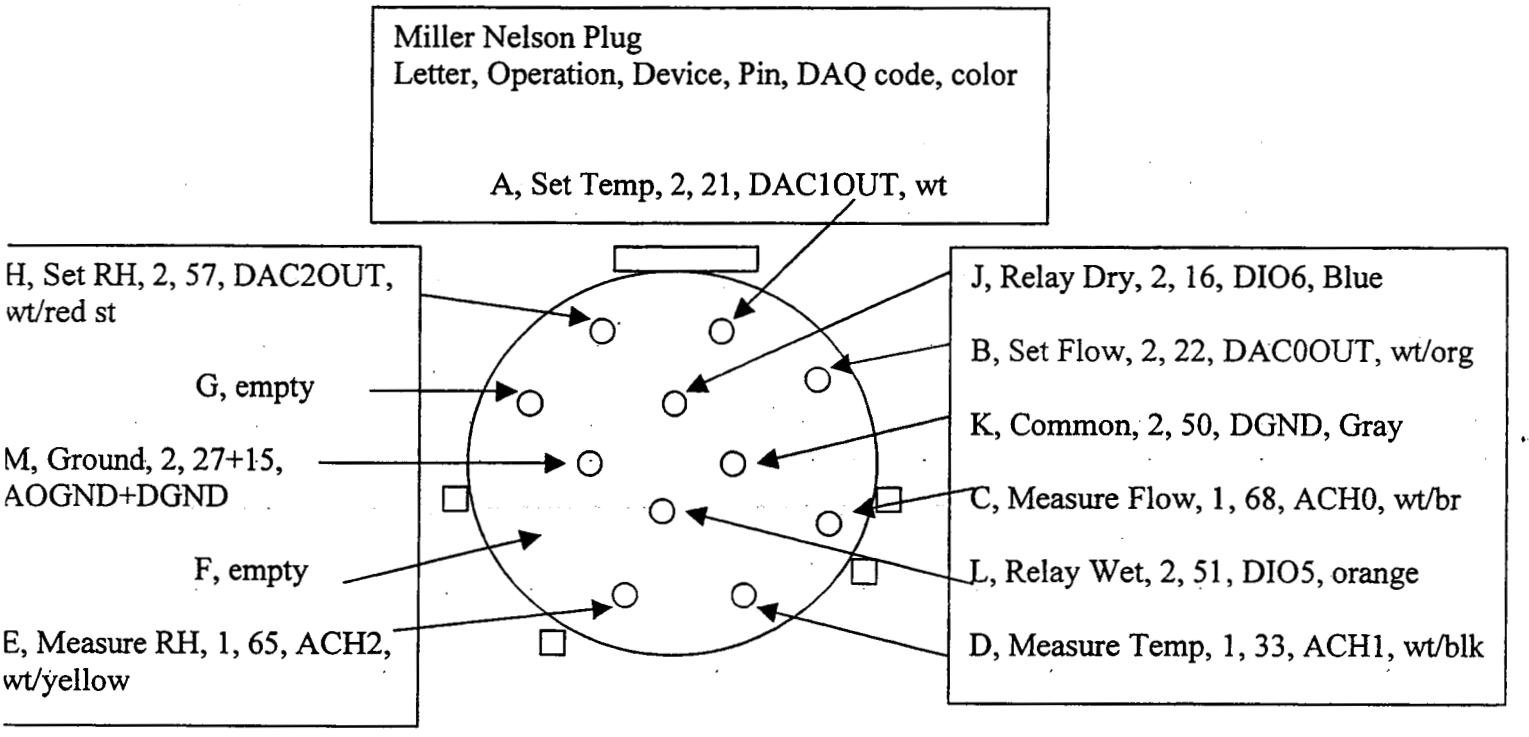
For both I would add more comments going to the front dialog box.

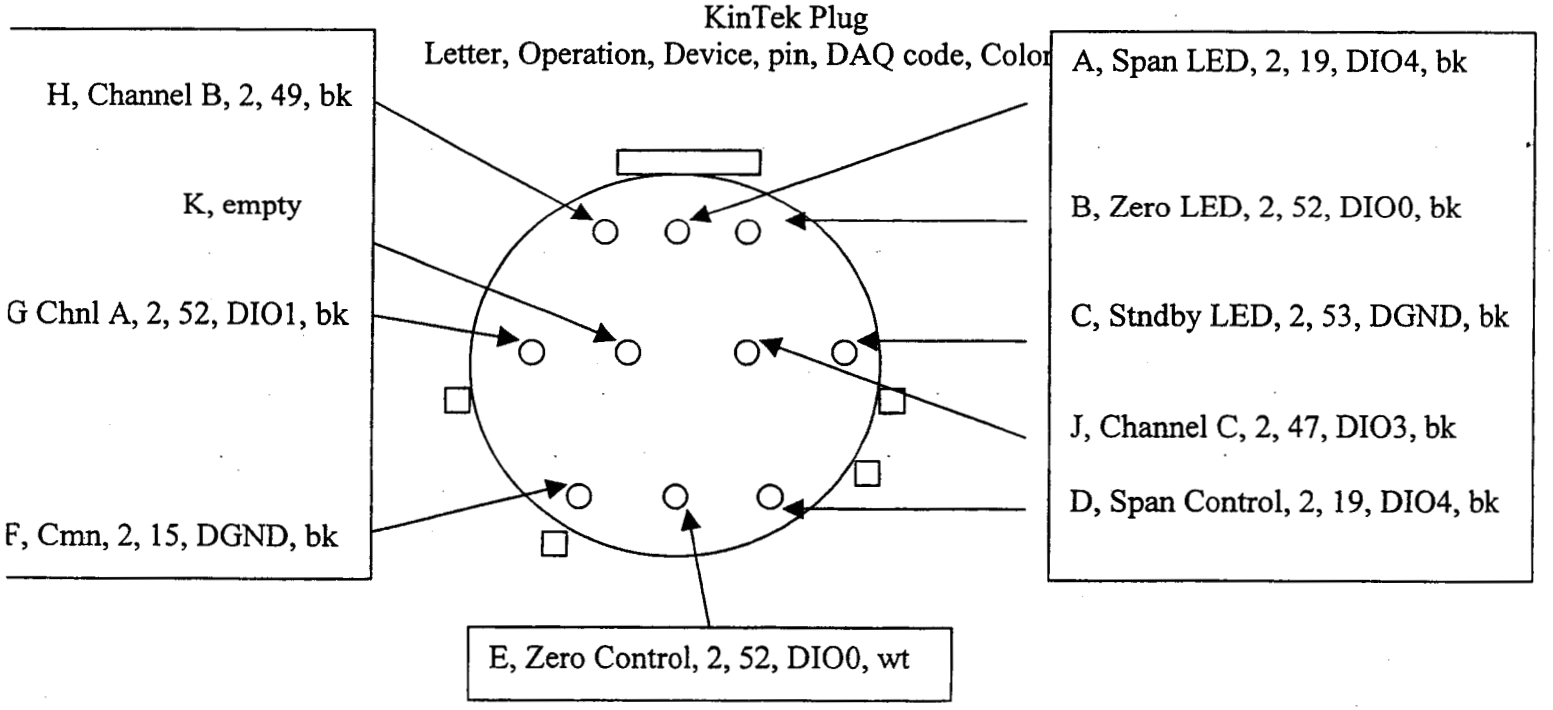
The relays for the MN wet/dry/split are 24 volt actuated. The LEDs for the KT zero/span/standby start at 53 volts. The rest of the controls for KT start at 3 volts. All these need relays to accurate. It would be more straightforward to have these in the machine than in an external relay system. The external system may be less expensive. Perhaps KT and MN could refurbish the machines to handle modern computer controls.

Outputs from the MN are available. There should be a way to use that and give a control saying change this parameter and wait until it stabilizes.

The control of the equipment becomes strange and wanders around when data is not being given to the instruments. A first fix would be to create a continuous loop with just inputs for the MN and KT values and outputs to those instruments. Run this program get things stable, then quickly shut it down and start the run script. A better fix would be the one mentioned above about waiting until stable. This may be related to not knowing the size of the array before hand.

It would be nice to wait until a user defined time of day to start.


Miller Nelson Plug
Letter, Operation, Device, Pin, DAQ code, color
A, Set Temp, 2, 21, DAC1OUT, wt
H, Set RH, 2, 57, DAC2OUT, wt/red st
G, empty
M, Ground, 2, 27+15, AOGND+DGND
F, empty
E, Measure RH, 1, 65, ACH2, wt/yellow
J, Relay Dry, 2, 16, DIO6, Blue
B, Set Flow, 2, 22, DAC0OUT, wt/org
K, Common, 2, 50, DGND, Gray
C, Measure Flow, 1, 68, ACH0, wt/br
L, Relay Wet, 2, 51, DIO5, orange
D, Measure Temp, 1, 33, ACH1, wt/blk


KinTek Plug
Letter, Operation, Device, pin, DAQ code, Color
H, Channel B, 2, 49, bk
K, empty
G Chnl A, 2, 52, DIO1, bk
F, Cmn, 2, 15, DGND, bk
A, Span LED, 2, 19, DIO4, bk
B, Zero LED, 2, 52, DIO0, bk
C, Stndby LED, 2, 53, DGND, bk
J, Channel C, 2, 47, DIO3, bk
D, Span Control, 2, 19, DIO4, bk
E, Zero Control, 2, 52, DIO0, wt