UNIVERSAL LAB INTERFACE

Software Developer's Guide

Distributed by:
Vernier Software
8565 SW Beaverton-Hillsdale Hwy.
Portland, OR 97225-2429
(503) 297-5317
FAX (503) 297-1760
dvernier@vernier.com

Version 5/8/96

ULI Software Developer's Guide

TABLE OF CONTENTS

About the ULI Software Developer's Guide

The Universal Lab Interface can be used with virtually any computer that has a serial communication port. Commercial programs for use with the ULI are available for Macintosh and IBM-compatible computers. This manual explains everything you need to know to control the ULI. It also describes how the ULI transfers data back to the computer. You can use this information to write your own programs. You can also use it to control the ULI with a telecommunications program. This manual does not make any assumptions about the type of computer you are using except in the examples in Appendix D.

This book has two major divisions. The first part is meant to be a quick overview, concentrating on the most frequently used features of the ULI. The second part is a reference section, listing information on every ULI feature.

To date, two versions of the ULI are in existence. The original ULI has a circuit board with all electrical components visible through a clear plastic cover. The original ULI has a number of different types of connectors, including two 5-pin DIN connectors labeled DIN1 and DIN2. It also has ten RCA jacks and three RJ-11 connectors labeled Port 1, Port 2 and AUX.

The second version of the interface, the ULI_{II}, was introduced in January 1995. It is a beige plastic box (electrical components are not visible). The ULI_{II} has four 5-pin DIN connectors, two RJ-11 connectors, and two stereo phone plugs.

Terms and symbols used in this manual:

- Throughout this manual, the term “original ULI” means the pre-1995 Universal Lab Interface with the 10 RCA jacks. “ULI_{II}” means the Universal Lab Interface with the 4 DIN connectors, and “ULI” means either one.
- <CR> for “carriage return” represents a press of the <Return> or <Enter> key.
- Keyboard combinations are indicated as shown in this example: <Ctrl>+C means press the <C> key while at the same time holding down the <Control> key.
- Numbers in hexadecimal notation are written with an “h” following them; for example, FFh represents FF in hexadecimal notation or 255 in decimal notation.
- When a range of values for a variable is specified, the values are specified in hexadecimal format using the following notation: <lowest possible: highest possible>. For example, the range of a one-byte number is <0h: FFh>.
- The names of registers and switches used with the ULI are enclosed in brackets. For example the ULI Time base register is labeled [T].
- All examples in this document are presented as terminal sessions and are printed in the Courier font. An example of this style is:

```
ULI Rev. 5.00
H4/3>
```

The ULI Software Developer's Guide is a collection of information gathered from a number of people, including Ron Budworth who designed the ULI and wrote the EPROM code, programmers who have written ULI applications, and instructors who have used the ULI in their classes. The information for this manual was collected and organized by David Gardner. Pat Cooney helped write the earlier version of this manual. Dave Vernier, Stephen Beardslee and Ron Budworth edited and verified this information. If you find errors or have suggestions for improving this manual, please contact Vernier Software.

I. INTRODUCING THE ULI

Overview

The Universal Lab Interface is really a microcomputer containing its own microprocessor, RAM, EPROM, and serial communications port. Many data collection routines are programmed into the EPROM of the ULI. This greatly simplifies the process of collecting data with the computer. To begin collecting data with the computer, you need only set up the appropriate connections for serial communication and enter relatively simple commands.

The information in this manual can be used in one of two ways:

1) You can use a telecommunications program to “manually” control the ULI. Data can be collected by typing just a few characters. Examples of this method appear throughout this manual. Some instructors use this procedure to collect data in a terminal program and then copy the data into a spreadsheet for analysis.
2) You can incorporate the commands used to control the ULI into your programs. Your software can then control and read a wide range of sensors and probes. Examples of such programs are presented in Appendix E.

Connections for Serial Communications

The ULI is designed to connect to virtually any computer or terminal that is equipped for serial communication. Two examples are the serial communications ports found on IBM compatibles which use the RS-232C standard and those found on the Apple Macintosh which use the RS-422 standard.

To connect the ULI to the serial port, you use the same cable that you would for an external modem.[1] Any standard modem cable that ends in a standard 25-pin male D connector should work. The absolute minimum cable that can be used connects pins 2, 3 and 7 of an RS-232C cable with 25-pin connectors on both ends. For IBM-compatible computers, this is a straight-through (pin-for-pin) cable.

On the ULI, pin 4 is connected to pin 5, and pins 6, 8, and 20 are wired together to maintain compatibility with the host computer or terminal. These pins are not used by the ULI.

Getting Started

Even if your primary goal is to write your own software to operate the ULI, we recommend that you use your computer as a terminal to try out all of the modes of the ULI before you start programming. When you are ready to start writing code, you will have a better understanding of how your programs should interpret the data stream from the ULI.

You will need a telecommunications program that lets your computer act like an on-line terminal. Any program that is normally used with a modem will have this capability. An example of a program in the Windows environment would be Terminal,which is included with Windows. An example of a terminal program in the MS-DOS environment would be ProComm by DataStorm. An example of a terminal program in the Macintosh Environment would be White Night by the Freesoft Company.

Verifying the ULI is Connected Properly and Working

1. Make sure the “Power Switch” is in the **OFF** position before applying power to the ULI. Plug the power adapter into the ULI and then plug the adapter into the wall.
2. Connect your computer to the ULI with the appropriate cable, securing the cable at both ends.
3. Run the telecommunications program on your computer. The communication protocol for the ULI is as follows:

---

[1] As far as communication protocol is concerned, the ULI looks like a Data Communication Equipment or DCE and as such, uses the same cable as a modem. The Transmit Data and Receive Data lines should **not** be crossed, as is the case in some cables.

Baud Rate:	300 to 38,400 baud (any rate in this range will work)
Word Length:	8 bits (8 data bits)
Parity:	None
Stop Bits:	1
Handshake:	X-on/X-off (<Ctrl>+Q/<Ctrl>+S)

Make sure your terminal emulation computer software is set to match these specifications. Baud rate is not a problem; the ULI will match the baud rate that you choose. In addition, you will probably want to set “Local Echo” (sometimes referred to as “Half Duplex”) so that you can see what you are typing. The ULI does not echo input data. Make sure that you are “on-line.” You may need to select “connect” or “on-line” when using some communications programs

4. Push the Power Switch to **ON**. If everything is connected properly, both LEDs on the ULI should light. The green LED indicates that power is being supplied to the ULI. The red LED is a status indicator that will turn on and off indicating that the ULI is doing something.
5. Press the <SPACE BAR>. A message similar to the following should appear on your screen:[2]

```
ULI Rev. 5.00
H4/3>
```

If this message does not appear, refer to Appendix B: Troubleshooting Guide. The H4/3> on the screen is the ULI *prompt*. It indicates that the ULI is waiting for the next command. You will see this often.

> Do the symbols in the ULI prompt mean anything? Yes, each of the first four characters has a particular meaning:
>
> - The first character indicates the display mode: H: Hexadecimal, B: Binary, or D: Decimal.
> - The second character indicates the number of bytes in the internal data accumulator, denoted [C] in this manual. The range is <0 : 4>.
> - The third character is either “/”, “\”, “!” or “*” indicating how the memory buffer in the ULI is being used.
> - The fourth character is the active I/O port (register [S]). The range is <0 : 3>.
>
> Later in this manual you will learn what each of these parameters means and how to change them.

When you pressed the <SPACE BAR>, the ULI used the signal it received from your computer to determine the baud rate. It then matches its baud rate to that of your computer.

6. Type “MFF” on your keyboard and then press <CR>.[3] The word “Data:” should appear on your screen and the red LED on the ULI should start flashing (approximately 1 sec. on, 1 sec. off).

If the LED is flashing and you have gotten the messages from the ULI on the screen, you can assume that everything is hooked up properly and that the ULI is functioning. The ULI will continue in this “self-test” mode until it receives a <Ctrl>+C from your keyboard, at which time the prompt will be displayed and the ULI waits for additional commands.

A Simple Example of Using the ULI to Collect Data

Just to show you how easy it is to collect useful data with the ULI, try the following:

Connect the ULI to the computer's serial port and start up a telecommunications program. Set the parity (none), data bits (8) and stop bits (1). Make sure the program is set for “half duplex” and “on-line” communication.

1. Turn on the ULI and press the <SPACE BAR>. The ULI prompt should appear.
2. Type “T000F42<CR>” (The “0” characters are zeros). This command sets the time base register [T] which controls the frequency of data collection. The value 000F42h sets the time base so that the ULI takes one reading per second.

---

[2]The exact prompt displayed on the screen depends on the ULI. If the ULI is an original ULI, the prompt will be ULI Rev. X.XX where X.XX is the version of the EPROM. The prompt for the ULI_{II} is ULI2 Rev. 1.00.

[3]You can type either MFF or mff.

3. Type “D<CR>”. This sets the ULI output to display decimal values.
4. If available, connect an analog sensor (temperature probe, pressure probe, etc.) to DIN1. If you do not have a sensor, connect the voltage test leads that came with the ULI to a 1.5-volt cell, red lead attached to the positive terminal and the black lead to the negative terminal.
5. Type “M8<CR>”. The ULI will respond with counts representing the voltage applied to DIN1. The number should vary as the voltage varies. If you are measuring the voltage of a 1.5-volt cell, it should remain constant. The second number displayed in each row will be meaningless, because no sensor is connected to DIN2 and the input is floating.
6. Press <Ctrl>+C to stop the data collection; that is, press the <C> key while at the same time holding down the <Control> key.

This example is typical of how the ULI is used: First, some registers and switches are set to control how data are collected and displayed, the appropriate data collection mode (for the type of sensor used) is started, and then data collection is ended.

ULI Registers

The ULI uses internal programs for all its modes. As in other computer programs, variables are used. In the ULI, the variables are called *registers* that determine how these programs run. These registers specify everything from how often to take samples to how the output is presented. The ULI registers can be read and written to using the same basic command. The most often used registers are described below:

Register T - Set Timer Period

The [T] register controls the timer period. In most of the data collection modes, this sets the time between sensor readings. The [T] register can be set to any value in the range 000000h to FFFFFFh. In most cases, the time between readings is the [T] register setting times 256 microseconds. Some useful settings (in hexadecimal) are listed below:

[T] setting	time (s)
000027	0.01
000187	0.1
00030D	0.2
0007A1	0.5
000F42	1.0
001E84	2.0
009896	10.0
039387	60.0

A complete table of the [T] register values are in Appendix E.

To read the [T] register, simply type the letter T followed by <CR>. The ULI will return the value.

To set the value of the [T] register, type the letter T followed by a six-digit hexadecimal number and then <CR>. Six digits must always be entered, so you use commands like:

```
T000F42
```

When the ULI is first turned on, the [T] register is set to 000000h, which is treated as 1000000h. This leads to the slowest possible sampling of once every 1.19 hours.

Example:

In this example, the [T] register is checked and is originally set to 000000. A [T] command is used to set it to 27h for sampling about every 0.01 seconds.

```
ULI Rev. 5.20
H4/3>T
000000
H4/3>t000027
```

```
H4/3>
```

Register S - Selects Active Ports

The [S] register selects the analog input ports (DIN ports or telephone-style connectors) to be used for data collection in many modes. The most frequently used [S] register settings are:

1 only port 1 active
2 only port 2 active
3 both ports 1 and 2 active

The value of the [S] register is always reported as character 4 of the ULI prompt.

To set the value of the [S] register, type the letter S followed by a number and then <CR>. When the ULI is first turned on, the [S] register is set to 3.

Register C - Data Word Length

The [C] register sets the size of internal data accumulators used by the ULI. These data accumulators can be 1 to 4 bytes. If the accumulator is set to 1 byte wide, it can hold numbers from 0 to 255 (FFh). It will overflow if it counts above FFh. If the [C] register is set for 4, the accumulator is 4 bytes wide and it can count up to FFFFFFFFh (2^{32}) before overflowing. The [C] register is set to 4 when the ULI is first turned.

The setting of the [C] register is always reported as character 2 of the ULI prompt.

To change the value of the [C] register, type the letter C followed by a number, and then <CR>.

ULI Commands

You can send commands to the ULI to set change the way data are displayed, to start data collection and to stop data collection. In most cases the ULI is not case sensitive.[4] Most commands you type while you are using the ULI can be either upper case or lower case.

Output Format: Hex/Binary/Decimal Display

The data reported by the ULI can be displayed in several different formats. Three commands are used to set the display format:

H (for hexadecimal)
D (for decimal)
B (for binary)

The (H)ex mode translates each data byte transmitted by the ULI into two printable ASCII characters (0-9 and A-F) with no delimiters. The decimal mode outputs data as decimal values separated by delimiters. Using the (D)ecimal command, you also can specify the character used as the delimiter between values and the number of complete records displayed on one line. The (B)inary setting outputs data as one or more 8-bit bytes with no delimiters. This setting can be useful when ULI values are being read into a computer program. It is not useful when you are using a ULI with a telecommunications program, since the data sent to the computer will often not be standard ASCII characters and will appear as strange characters on the screen.

If the display switch is set for Hex or Decimal, each output string is terminated with a <CR><LF>. With the Binary setting, no <CR><LF> is added.

The H and B Commands

To change the ULI output display format use the commands described below:

Command: H<CR> (to select (H)exadecimal, ASCII)

[4]Exceptions are the escape codes used to control the Digital Out lines and the Alpha commands in Mode 6.

B<CR> (to select (B)inary)

The D Command

The decimal display format has several options, so the commands to control and check its status are more complicated.

Command: D<CR> (to select Decimal display)

This command sets the ULI output format switch for Decimal display and returns the ASCII code of the delimiter and the number of complete records per line.

Reporting Format: aabb

where aa is the ASCII code of the delimiter used between fields of the output and bb is the number of complete records per line. These return values will always be expressed in hexadecimal notation.

If you turn on and initiate contact with the ULI and immediately issue the D command, you will see the following displayed on the screen:

2C01

This means that a comma (ASCII 2Ch) is set for the delimiter and one complete record will be displayed per line.

You can change the delimiter or the number of complete records per line by using the D command as shown below:

Command: Daabb<CR>

where: D specifies the decimal mode
aa = the hex code for the desired delimiter <00:FFh>
bb = the number of complete records to be displayed per line <01:FFh>

Example: If you enter D0901<CR>, you are specifying that each decimal output value should be separated by a tab (ASCII code 09h) and only one complete set of measurements from the ULI should be displayed per line on the screen.

The table below lists the ASCII codes of some characters that are commonly used as delimiters.

Character	**ASCII Code (Decimal)**	**ASCII Code (Hex)**	**Character name**
<tab>	09	09	tab
,	44	2C	comma
<space>	32	20	space
<cr>	13	0D	return or enter

<Ctrl>+C

<Ctrl>+C is used to tell the ULI to stop whatever it is doing and get ready to accept more commands. Generally, <Ctrl>+C is used to stop a data collection mode. On some computers, you may find you will also have to hold down the <Shift>key at the same time.

Other commands

The ULI also has several other commands. These commands are generally used to set the output of the ULI. The control commands set the data stream from the ULI. The escape commands turn the digital output lines on or off. In most cases the ULI is not case sensitive. Most commands you type while you are using the ULI can be either upper case or lower case. The escape commands listed here are exceptions, and they must be executed with upper case letters.

Commands to interrupt and resume communications:

<Ctrl>+S	interrupts data transmission
<Ctrl>+Q	resumes data transmission that was stopped with <Ctrl>+S

Commands to control output lines:

<ESC>A	sets digital output 1 high
<ESC>B	sets digital output 1 low
<ESC>C	sets digital output 2 high
<ESC>D	sets digital output 2 low

Commands to control how the ULI memory is used:

R	specifies the number of pages of RAM to be reserved for the buffer <01 : 20h>
/	selects normal (ring) buffer operation
\	disables the ring buffer
!	selects flash mode, collecting one RAM full of data and transmitting when done
*	selects the burst mode, collecting RAM full of data and transmitting, then repeating

Commands controlling triggering:

I	sets the digital and analog triggering conditions
V	sets the analog voltage triggering levels

Other commands:

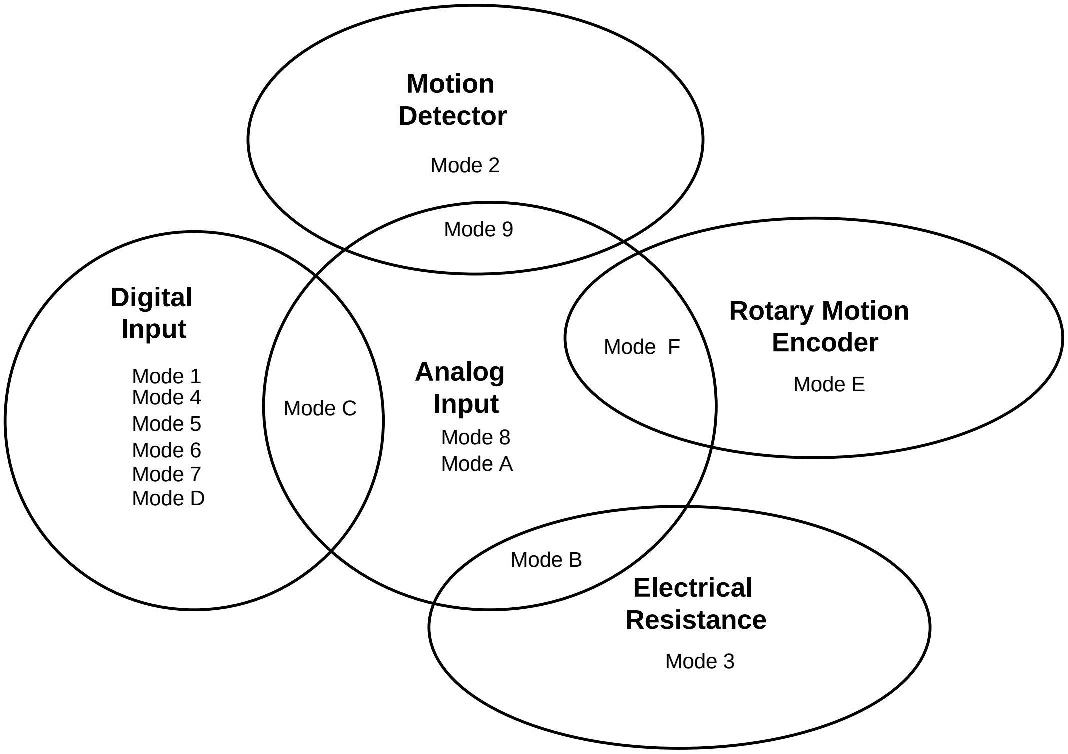
E	sets the time base <01 : FFh>; 00h means 100h which equals 256 µs
J	sets the mask for error trapping

ULI Data Collection Modes

When you take data using the ULI, built-in routines in the ULI EPROM do most of the work. All you usually have to do is set a few parameters, such as how often data should be collected, and send the command to start the appropriate data collection mode. Currently, the ULI supports 15 different data collection modes. The data collection modes are started by sending the “M” character followed by a hexadecimal value representing the mode. The following table lists the data collection modes that are currently available.

Mode 1	reports the time at which any digital input port changes its state
Mode 2	determines echo times from the motion detector
Mode 3	reports electrical resistance (used on the original ULI only)
Mode 4	reports period of the signal on digital input 1
Mode 5	reports each complete pulse width (in µs) at one digital input
Mode 6	stopwatch emulation
Mode 7	reports the pulse count at one digital input during each preset timer interval
Mode 8	reports, after each preset timer interval, values of one or both voltage inputs
Mode 9	runs the Motion Detector and an analog probe at the same time
Mode A	reads up to 14 analog voltages once each preset time interval
Mode B	resistance and analog readings
Mode C	reports the status of the digital outputs and the analog voltages inputs
Mode D	sequential pulse period timer
Mode E	reads rotary motion detector
Mode F	reads rotary motion detector and analog inputs

The heart of the ULI’s data collection mode is analog data collection, as the bubble diagram below shows. The center circle represents analog data collection. Modes 8 and A are designed specifically for collecting only analog data. Many other modes combine analog data collection with some other type of data collection.



Of the many data collection modes used with the ULI, we will go through the three most often used ones below.

Mode 8 - Analog Ports 1 and 2 only

Mode 8 reads one or two “voltage type” analog sensors (temperature, force, pH, etc.) plugged into the DIN connectors, or the modular phone jacks. Voltage readings range from 0 to 1023 (3ffh) on the original ULI, and 0 to 4095 (0FFFh) on the ULI_{II}. Each count has the value of 0.005 volts for the original ULI, and 0.00125 volts for the UL_{II}. In most cases these raw counts will be converted to more meaningful values (temperatures, forces, etc.) using a conversion equation.

Before using this mode, set the sample period using the [T] register and make sure the output display is set to decimal, hexadecimal, or binary as you want to receive the data. Type M8 and then <CR> to start analog data collection.

In this example, two analog voltages are fed into DIN1 and DIN2. They are read at the rate of 10 per second and displayed in decimal format.

```
ULI2 Rev. 1.00
H4/3>t000187
H4/3>d
2C01
D4/3>m8
Data:
392,177
431,195
204,92
73,34
87,41
309,140
390,176

D4/3>
```

Mode 2 - Distance Reading

This mode was designed to operate a Motion Detector connected to Port 2. When this command is received, the ULI will cause a burst of ultrasonic sound waves to be produced by the Motion Detector and start a timer. The ULI then will wait for the echo of the ultrasound. When the echo is received, the timer is stopped and the time is reported. The time is measured in microseconds. Knowing the speed of sound in air (about 343 m/s), you can calculate the distance to the object easily.

In this example, the time between readings is set to 0.1 seconds. The Motion Detector is aimed a wall about 2 meters away and data are collected. The output display is left in the hexadecimal mode. The average value returned is the time a pulse from the Motion Detector took to travel to the wall and back. The average is 2B35 hexadecimal (11061 decimal). The time of a one way trip is then 5.53 milliseconds. If the speed of sound in air is 343 m/s, this time represents 1.89 meters.

```
ULI2 Rev. 1.00
H4/3>t000187
H4/3>m2
Data:
2B36
2B34
2B34
2B35
2B35
2B35
2B34
2B34

H4/3>
```

Mode 1 - Simple Interval Timing

This mode was designed to measure time between status changes on the digital lines. In particular, this was meant as a way to collect times for photogate timing experiments.

This mode starts an internal timer running at 1 microsecond per tick. It then samples the two digital input ports and reports the time when either of the digital input lines change status. A character indicating the status of the inputs is also reported. Once the mode is started, the time in microseconds is reported each time the status of either digital input changes. This process continues until the mode is terminated (<Ctrl>+C) or an error.

The only setup required for this data collection mode is to decide what output format will be used and how many bits of precision will be displayed (using the [C] register.

The times collected in this mode have a resolution of about 1 µs. The first value reported by the ULI is an indication of how much time is required to check the digital lines, and is a measure of the accuracy of the timing.

The output in the decimal mode is {# of microseconds},{status of inputs}. The status of the inputs can be:

0 - both inputs low
1 - input 0 high, input 1 low
2 - input 0 low, input 1 high
3 - both inputs high

In this example, a terminal program is used with a ULI_{II}. A command is then issued to the ULI to display data in decimal format. Mode 1 is initiated and four opaque bars are passed through the photogate attached to Digital input number 1. Later a <Ctrl>+C was issued and the timing mode is terminated. Again, the ULI is put into Mode 1 and time data collected for four opaque bars passing through a photogate connected to DG2.

```
ULI2 Rev. 1.00
H4/3>d
2C01
D4/3>m1
Data:
17,1
```

```
2563954,2
2578553,3
2591402,2
2606694,3
2615972,2
2627835,3
2641626,2

D4/3>
```

II. REFERENCE SECTION

ULI Data Collection Modes

The following information is intended as a reference section. This is where all of the gory details about each collection mode and command are listed. An example is provided for each mode/command, which should help you to understand the text. The ULI's data logging modes collect data from triggered analog data to encoded digital data. With the exception of modes 0 and FF, all data collection modes have decimal, hexadecimal and binary output capabilities.

When writing programs that read data displayed in binary form, each mode will first send the character string "Data:" followed by the <CR><LF> characters and then send the data stream. Some modes, however, send a stream of test data before the actual data.

Once data collection modes are initiated, they can be terminated by issuing a <Ctrl>+C command. This command is case sensitive and you may find that a <Ctrl><Shift>C must be issued from a terminal program to interrupt data collection.

In all of the examples presented that use the [E] and [T] registers, the [E] register will be set to FAh, which sets the time base to 250 microseconds. The [T] register will be set precisely for "exact" times, which are multiples of 250 microseconds.

Mode 0 - Reset ULI

Description:

Mode 0 resets the ULI to its initial startup condition. All flags and registers are set as they are when the ULI is first turned on:

Display mode is set to hexadecimal format - [H]
Four internal data accumulators are used - [C] = 4
Normal buffer use is selected - [/]
The Analog In lines of both DIN connectors will be read - [S] = 3
Both internal timers are set to zero - [E] = 0 and [T] = 0

The ULI does not return any characters when this command is executed. Immediately after issuing the M0 command, the user should press the <SPACE BAR>. This allows the ULI to determine the baud rate being used. At this time, the standard ULI prompt will be displayed.

This mode can be used when you want to make sure the ULI is in its original startup condition.

Setup:

The only setup required for this mode is to issue it from an active session. This command will not work if the application gets stuck in a collection mode at one baud rate and then tries to reattach at a different baud rate.

Sending a BREAK[5] at any time also resets the ULI.

Output:

This mode has no output until a <Space> character is sent to the ULI.

Example:

```
...
H4 /3>m0
ULI Rev. 5.20
H4/3>
...
```

[5] A BREAK is a state of the communications chip in the computer. This state sends a stream of 0's as the signal to the ULI. In many development systems, the asynchronous communications library includes SetBreak and ClearBreak methods.

Mode 1 - Simple Interval Timing

Description:

This mode was designed to measure time between status changes on the digital lines. In particular, this was meant as a way to collect times for a photogate timing experiments.

This mode starts an internal timer running at 1 microsecond per tick. It then samples the two digital input ports and reports the time when either of the digital input lines change status. A character indicating the status of the inputs is also reported.

Once the mode is started, the time accumulator is cleared and the initial status of the digital inputs is determined. Once done, the current time accumulator reading and initial status of the digital inputs is placed in the output buffer. During the next iteration of the program loop, the digital inputs are again sampled and the current status compared to the previous status. If a difference is detected, a “snapshot” of the Time Accumulator is again taken and this and the latest digital input sample is placed in the output buffer. This process continues until the mode is terminated (<Ctrl>+C) or an error.

Ports Used:

Mode 1 reads the status of the digital inputs. These inputs are available on DG1 and DG2, respectively. These inputs are also available on pin 6 of Port 1 and pin 6 of Port 2, respectively.

Setup:

The only setup required for this data collection mode is to decide what output format will be used and how many bytes of precision will be displayed. Exception handling may also be set.

Associated Commands:

- Use the [B], [D] and [H] registers to set the display mode.
- Use the [C] command to set the number of bytes used for counting. This also sets the value at which the counters overflow. For example, if [C] is set to 1, the counters will overflow at 255 or 255 µs. If [C] is set to 4, the counters overflow after 4294967295 µs or about 1.19 hours later.

Exception Handling:

Bit #4 of the Error Register is set when the time counters rolls over. Setting the matching bit in the Error Mask Register will cause Mode 1 to terminate when this condition occurs.

Display Options:

Data may be reported in hexadecimal, binary, or decimal format.

When the hexadecimal and binary display modes are used, the times displayed follow the overflow rule set by the [C] register. When the decimal display mode is chosen, the times are display as if the [C] register were set to 4.

Output:

The times collected in this mode have a resolution of about 1 µs. The accuracy of the measurements is about 17 µs. The resolution figure is always the first data point reported by the ULI. The first value reported by the ULI is an indication of how much time is required to check the digital lines, to check the timers, to place all of this information into the communications buffer and to report the data.

The output in the decimal mode is {# of microseconds},{status of inputs}. The status of the inputs can be:

0 - both inputs low
1 - input 0 high, input 1 low
2 - input 0 low, input 1 high
3 - both inputs high

The output in the hexadecimal mode is {# of microseconds}{status of inputs}. The number of microseconds and the status of inputs field are **not** delimited. The status of the inputs is as stated above.

The output in the binary mode is equal to the value in the [C] register plus one. The last byte is used for communicating the status of the photogates. The data are not followed by a <CR><LF> so it is up to the driver program to synchronize the data.

Example:

In a terminal program, the ULI (a ULI_{II} in this case) is turned on and put into Mode 1. By default, the values are displayed in hexadecimal format. After passing four opaque bars through the photogate attached to DG1, a <Ctrl>+C was issued and the timing mode is terminated. A command is then issued to the ULI to display data in decimal format. Again, the ULI is put into Mode 1 and time data collected for four opaque bars passing through a photogate on DG2.

```
ULI2 Rev. 1.00
H4/3>m1
Data:
0000001102
002FEC5D03
00300BBC02
003032D203
0030503602
003063DC03
00307FD802

H4/3>d
2C01

D4/3>m1
Data:
17,1
2563954,3
2578553,1
2591402,3
2606694,1
2615972,3
2627835,1
2641626,3
2651634,1

D4/3>
```

Mode 2 - Distance Reading

Description:

This mode was designed specifically to operate a Polaroid Sonar system or Motion Detector. The Motion Detector must be connected to Port 2. When this command is received, the ULI will launch a ultrasonic burst from the Motion Detector and start a timer. It then will wait for the echo of the ultrasound. When the echo is received the timer is stopped and the time is reported. If no echo is received in the time interval specified by the [T] register, a “time out” value of FFFFh is reported.

Ports Used:

Port 2 only

Setup:

The only setup required for this collection mode is to set the sample period and to set the display mode. Setting the sample period also sets the maximum distance the motion detector can measure.

Associated Commands:

- Use the [B], [D] and [H] registers to set the display mode.
- The timer may be set (using the [C] register) to count using 2 to 4 bytes.
- The [T] and [E] registers are used to control the sample period.

The Motion Detector will not work properly if the sampling increment is set to less than about 0.01 seconds. Sometimes problems occur with sample times less than 0.02 seconds. Note also that as you make the time between readings smaller, you limit the maximum range of the Motion Detector. The echo may not have enough time to return before the next measurement starts. The table below gives an indication of maximum range vs. [T] setting (assuming [E] = 0):

[T]	time between readings	maximum range
000027	0.01 sec	1.72 meters
00004E	0.02 sec	3.44 meters
000075	0.03 sec	5.16 meters
00009C	0.04 sec	6.88 meters

Exception Handling:

Not Applicable

Display Options:

Data may be reported in hexadecimal, binary, or decimal format.

Output:

The time it takes for the echo to return is reported, measured in microseconds with a resolution of 1 µs.

Example:

In this example, the time between readings is set to 0.04 seconds. The motion detector is aimed a wall about 2 meters away and data are collected.

The average value returned is the time a pulse from the motion detector took to travel to the wall and back. The average is 2B35 hexadecimal (11061 decimal). The time of a one way trip is then 5.53 milliseconds. If the speed of sound in air is 343 m/s, this time represents 1.89 meters.

```
ULI2 Rev. 1.00
H4/3>efa
H4/3>t0000a0
H4/3>m2
Data:
2B36
2B34
2B34
2B35
2B35
2B35
2B34
2B34

H4/3>
```

Mode 3 - Electrical Resistance Reading

This mode is only applicable to the original ULI.

Description:

This mode uses resistive Input 1 or 2 to measure the time (in µs) taken to charge the ULI's internal 0.01µF timing capacitors. Resistive inputs 1 and 2 are both reported, independent of the value of the [S] register's setting. The minimum resistance that can be measured is about 1 k and the minimum value reported is 44 (2Ch) +/- 5%.

This mode is normally used for determining the resistance of sensors such as a thermistors or photoresistors.

The resistance to be measured should be connected between +5V and the resistive input line. Note that the outside of the resistive input RCA connectors is +5V.

Ports Used:

Resistive inputs

Setup:

The only setup required for this collection mode is to set the sample period and to set the display mode and precision

Associated Commands:

- Use the [B], [D] and [H] registers to set the display mode.
- The [E] and [T] registers are used to determine the sampling interval. The [T] register also determines the time period after which the mode will give up waiting for the capacitor to charge and report an overflow. Refer to the table of [T] settings to be used with this mode (*Register T - Set Timer Period*).
- The timer may be set (using the [C] register) to count using 2 to 4 bytes.

Exception Handling:

Not Applicable

Display Options:

Data may be reported in hexadecimal, binary, or decimal format.

Output:

Using decimal display, the count from both resistive inputs is displayed separated by a delimiter. Using the hexadecimal display, the data are displayed with 4 characters for each of the two inputs. The count is proportional to the resistance.

In the worst case, the resolution in the time reported for the capacitor to charge is about 10 µs.

Here are some typical results for this mode:

Resistance, k	Count, hexadecimal (decimal)
11.0	88 (136)
24.2	107 (263)
32.4	156 (342)
43.8	1C6 (454)

Example:

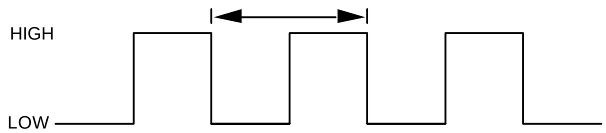
In this example, the ULI reads the resistive inputs every second and displays the elapsed time in decimal format. A 10-k variable resistor is connected to resistive input #1 and a 1 k, 10% carbon film resistor is attached to resistive input #2. During the sampling interval, the knob on the variable resistor is turned to different positions.

```
ULI2 Rev. 1.00
H4/3>t000fa0
H4/3>efa
H4/3>d
D4/3>m3
Data:
121,44
121,44
104,44
67,44
44,44
44,44
48,44
69,44
86,44
119,44
121,44
D4/3>
```

Mode 4 - Pulse Period Timing

Description:

This mode is normally used for determining the frequency of an oscillator or V/F converter. Mode 4 measures the time from falling edge to falling edge of the signal on Digital Input #1 and reports the period in microseconds. Only one period is measured and reported. To measure successive periods, Mode D should be used (described later).



Timing Using Digital Input #1

Ports Used:

Digital port 1 only

Setup:

The only setup required for this collection mode is to set the display mode and precision

Associated Commands:

- Use the [B], [D] and [H] registers to set the display mode.
- Use the [C] command to set the number of bytes used for counting. If [C] = 1 the timer overflows in 256 microseconds. If [C] = 4 the timer takes 4294967296 microseconds or 1.19 hours to overflow.

Exception Handling:

Bit #4 of the Error Register is set when the time counters rolls over. Setting the matching bit in the Error Mask Register will cause Mode 4 to terminate when this condition occurs. See the description of the [J] register for details.

Display Options:

Data may be reported in hexadecimal or decimal format.

Output:

The times collected in this mode have a resolution of about 1 µs. The data are displayed as a time in microseconds. Using hexadecimal display, the number of digits displayed depend on the display mode and the value of [C].

Example:

In this example, the ULI_{II} is used to measure the period of an oscillator that has a period of about 0.8 seconds.

Note the first value, it is well out of range. This will happen about half the time as the mode 4 program assumes the digital line is high before the timing starts. If the digital line is low when the program first checks, the mode 4 program assumes a high-to-low transition just occurred and starts timing. As a result, the first value returned by the ULI should not be considered to be valid.

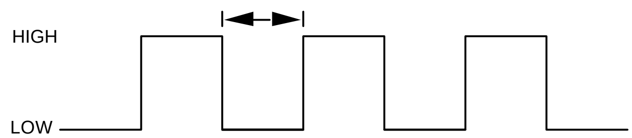
```
ULI2 Rev. 1.00
H4/3>m4
Data:
000798B4
000C47F1
000C483A
000C4798
000C488A
000C4769
000C481F
000C47EA
000C47B1
000C485D
000C47E0
```

```
H4/3>
```

Mode 5 - Pulse Duration Timing

Description:

Mode 5 monitors either digital input 1 or digital input 2 and reports the pulse width; that is, it times the interval during which the digital input line is either low or high. The period is reported in microseconds. This mode is normally used with photogates.



Timing Used in Mode 5

The time monitored by Digital Input 1 is programmable. If monitoring Digital Input 1 and register [I], bit 0 is set to 0, mode 5 will time how long the signal was LOW. If monitoring Digital Input 1 and register [I], bit 0 is set to 1, mode 5 will time how long the signal was HIGH. In other words, the status defined by bit 0 of the [I] register is what is measured when digital input 1 has the same status..

Mode 5 always times how long the status of Digital Input 2 is HIGH.

Ports Used:

Digital Input 1 and Digital Input 2

Setup:

The setup required for this collection mode is to set the sample period and to set the display mode and precision. You must also set the Digital Input line to be used. If you choose Digital Input 1, you must also choose which level duration is to be measured. Exception handling may also be set using the [I] register (see Exception handling section below).

Associated Commands:

- Use the [B], [D] and [H] registers to set the display mode.
- Use the [C] command to set the number of bytes used for counting. If too few internal accumulators are used to display the time, zero will be returned.
- Use the [S] register to select the Digital Input. Only one port can be used at a time and the ULI will act as if it were “locked-up” if the signal on the selected input does not change. The [S] register can have the value 1 which selects Digital Input 1 or the value 2 which selects Digital Input 2.
- Use the [I] register to select the level of Digital Input 1 to be timed.

Exception Handling:

Bit #4 of the Error Register [I] is set when the time counters rolls over. Setting the matching bit in the Error Mask Register will cause Mode 5 to terminate when this condition occurs. See the Register [I] section of this manual for details.

Display Options:

Data may be reported in hexadecimal or decimal format.

Output:

The times collected in this mode have a resolution of about 1 µs. The data are displayed as a time in microseconds. Using hexadecimal display, the number of digits displayed depend on the display mode and the value of [C]. If [C] = 1, the timer overflows in 256 microseconds. If [C] = 4, the timer takes 4294967296 microseconds or 1.19 hours to overflow.

Example:

In this example, the same oscillator used in the example of Mode 4 is used. First, digital line 1 is used to measure how long the signal is low. Then, digital line 2 is selected to measure how long the signal is high.

```
ULI2 Rev. 1.00
H4/3>s1
H4/1>i0
H4/1>m5
Data:
0003F43B
0003F43B
0003F43A
0003F452
0003F447
0003F459
0003F420

H4/1>s2
H4/2>m5
Data:
0008546D
0008543B
00085428
000854BF
000853FE
00085487
00085467

H4/2>
```

Mode 6 - Stopwatch Emulation

Description:

Mode 6 simulates a stopwatch timer using both digital inputs and the computer keyboard. Each time an event takes place, the stopwatch reports the elapsed time since it was started. An event can be either a transition on a digital input line or a key pressed by the user.

Unlike other modes, Mode 6 expects user interaction. Use the following keys to control the stopwatch:

G - Go - Start the stopwatch
S - Stop - Stop the stopwatch, resets stopwatch to zero
L - Lap - Request lap time

All stopwatch commands must be issued in upper case. This is one of the few times the ULI is case sensitive.

Ports Used:

Digital Ports 1 and 2

Setup:

The setup required for this collection mode is to set the sample period and to set the display mode and precision. Exception handling may also be set.

Associated Commands:

- Use the [B], [D] and [H] registers to set the display mode.
- Use the [C] command to set the number of bytes used for counting. If too few internal accumulators are used to display the time, zero will be returned.
- The overflow time is controlled by the [T] and [E] registers. The standard range of sample times is 256 µs to 1.19 hours.

Exception Handling:

Bit #4 of the Error Register is set when the time counters rolls over. Setting the matching bit in the Error Mask Register will cause Mode 6 to terminate when this condition occurs.

Display Options:

Data may be reported in hexadecimal or decimal format.

Output:

The times collected in this mode have a resolution of about 1 µs. The output has the following format:

A, BB

A = Status character "signature" - describes the event taking place:

"S"	- User has pressed S
"L"	- Uses has pressed L
"J"	- Port 1 has changed from high-to-low
"K"	- Port 2 has changed from high-to-low
"I"	- the preset timer has elapsed

BB = "time" event took place

Times are reported as the number of periods set by the [E] register (usually 256 µs). For example, [E] = 64h sets the clock period to 100 µs and [E] = FAh sets it to 250 µs. Until G is pressed, the time will be reported as zero.

Example:

This example demonstrates the ULI in timing mode 6 reading the events both before and after the timers are started. The Stopwatch Timing mode will monitor both digital inputs in addition to the "S", "L", "G" and <Ctrl> "C" keys. Times are displayed in decimal format. Notice the timer is reset to zero when the "S" key is pressed.

```
ULI2 Rev. 1.00
H4/3>D
2C01
D4/3>M6
Ready
G
L,16734
K,44124
J,50981
S,69101
K,0
J,0
L,0
G
K,6651
J,11542

D4/3>
```

Mode 7 - Event Counting

Description:

Mode 7 monitors a digital input for a specified amount of time and reports the total number of pulses. This mode is generally used for monitoring the frequency of square waves or pulses from radiation counters.

Ports Used:

Digital Ports 1 and 2

Setup:

The setup required for this mode is to set the sample time and to set the display mode and precision. The digital input must be selected. Exception handling may also be set.

Associated Commands:

- Use the [B], [D] and [H] registers to set the display mode.
- Use the [C] command to set the number of bytes used for counting. If too few internal accumulators are used to display the time, zero will be returned.
- Use the [S] register to select the digital input. Only one port should be used at a time. The [S] register should be set to 1 to select digital input 1 or set to 2 or three selects digital input 2.
- Use the [T] and [E] registers to set the sample period.

Exception Handling:

Bit #4 of the Error Register is set when the time counters rolls over. Setting the matching bit in the Error Mask Register will cause Mode 7 to terminate when this condition occurs.

Display Options:

Data may be reported in hexadecimal or decimal format.

Output:

The times collected in this mode have a resolution of about 1 µs. The data are displayed as a count. Using hexadecimal display, the number of digits displayed depends on the display mode and the value of [C].

Example:

In this example, a radiation counter is attached to digital input 2 (notice the prompt) and the sampling time is set for 10 seconds.

```
ULI2 Rev. 1.00
H4/3>efa
H4/3>t009c40
H4/3>d
2C01
D4/3>m7
Data:
407
443
444
417
418
400
420
432

D4/3>
```

Mode 8 - Analog Ports 1 and 2 only

Description:

Mode 8 reads any two “voltage type” analog sensors (Temperature, Force, pH, etc.) plugged into the DIN connectors, the modular phone jacks or, on the original ULI, the RCA connectors labeled “Voltage In”. Voltage readings range from 0 to 1023 (3ffh) on the original ULI, and 0 to 4095 (0FFFh) on the ULI_{II}. Each count has the value of 0.005 volts for the original ULI, and 0.00125 volts for the ULI_{II}.

Ports Used:

Port 1/DIN1 and Port 2/DIN2

Setup:

The setup required for this collection mode is to set the sample period and to set the display mode and precision. The I/O port to be used and the number of bytes used to report the values must also be selected.

Associated Commands:

- Use the [B], [D] and [H] registers to set the display mode.
- Use the [C] command to set the number of bytes used for reporting voltages
  [C]=1 sets the maximum reported value as 255
  [C]>1 sets the maximum reported value as 1023 on the original ULI and 4095 on the ULI_{II}
- Use the [S] register to select the Digital Input. Only one port should be used at a time. The [S] register should have the value 1 which selects Digital Input 1 or the value 2 which selects Digital Input 2.
  [S]=1 selects Voltage Input 1
  [S]=2 selects Voltage Input 2
  [S]=3 selects both Voltage Input 1 and 2
- Use the [T] and [E] registers to set the sample period.
- Use the Flash [!] or Burst [*] commands for extra fast data collection.
- Use the [R] register to set the number of buffer pages.
- Use the [I] and [V] registers to set triggering.

Exception Handling:

Not Applicable

Display Options:

Data is in a two-byte, 10-bit format from the original ULI (12-bit from the ULI_{II}) if the [C] register is set > 1, or in one-byte, 8-bit format if [C]=1. The 8-bit data format is useful when you are trying to obtain the very fastest sampling possible. It has a resolution of 1 part in 256 or 20 mV.

Output:

Data may be reported in hexadecimal or decimal format. If both Port 1 and Port 2 are being used, they are listed one after the other and if displayed in decimal, separated by the delimiter.

Sampling skew (time difference between sampling port 1 and port 2) is approximately 1 s when both ports are sampled.

Example:

In this example, the analog voltages fed into DIN1 and DIN2 are read at an exact rate of 10 per second and displayed in hexadecimal format. The same example is then repeated but with the values displayed in decimal format.

```
ULI2 Rev. 1.00
H4/3>efa
H4/3>t000190
H4/3> m8
Data:
016600A2
016600A2
019800B8
01AA00C0
002E0016
00D40060
016200A0

H4/3>d
2C01
D4/3>m8
Data:
392,177
431,195
204,92
73,34
87,41
```

```
309,140
390,176

D4/3>
```

Mode 9 - Distance and Analog Reading

Description:

This mode is specifically designed for use with an analog sensor (usually a force sensor) on Port 1 and the Motion Detector on Port 2. Measurements are taken at user-selected sample intervals. When this mode is started, the ULI will launch a ultrasonic burst from the Motion Detector and start a timer. It then will wait for the echo of the ultrasound. A sample on the analog sensor is then taken and the ULI waits for an echo pulse from the motion detector. When the echo is received the timer is stopped and the time is reported. If no echo is received in the time interval specified by the [T] register, a “time out” value of FFFFh is reported.

Ports Used:

Port 1 (Analog) and Port 2 (Motion Detector)

This mode is most often used with a force sensor connected to Port 1.

Setup:

The only setup required for this collection mode is to set the sample period and to set the display mode.

The [A] command will cause the ULI Force Probe electronics to lose “zero.” It is also possible that future modes and/or commands will also affect this adjustment. For this reason it is advisable to use the following sequence when using the ULI Force Probe:

A<CR>	[Get the ID codes for the Force Probe (Port 1) & Motion Detector (Port 2) and check for validity]
Tnnnnnn<CR>	[Set the timer]
S1<CR>	[Set the input to refer to the Force Probe only]
P400<CR>	[Zero the Force Probe]
M9<CR>	[Take readings]

Associated Commands:

- The [B], [D] and [H] registers are used to set the display mode.
- The [T] and [E] registers are used to control the sample period. Refer to Appendix D for a table of [T] and [E] settings.

  The Motion Detector will not work properly if the sampling increment is set to less than about 0.01 seconds. Sometimes problems occur with sample times less than 0.02 seconds. Note also that as you make the time between readings smaller, you limit the maximum range of the Motion Detector. The echo may not have enough time to return before the next measurement starts. The table below gives an indication of maximum range vs. [T] setting (assuming [E] = 0):

[T]	time between readings	maximum range
000027	0.01 sec	1.72 meters
00004E	0.02 sec	3.44 meters
000075	0.03 sec	5.16 meters
00009C	0.04 sec	6.88 meters

- Use the [P] register to zero the force probe.
- The [S] and [C] registers are ignored in this mode.
- Use the [A] command to verify that the force probe and motion detectors are in the correct ports.

Exception Handling:

Not Applicable

Display Options:

Data may be reported in hexadecimal, binary, or decimal format.

Output:

The time measured by the Motion Detector is reported first, followed by the Analog sensor reading. If they are displayed in decimal, the two numbers are separated by the delimiter.

If the ULI does not find a Motion Detector it will “hang” until a <Ctrl>+C is received. If no echo is received within 2^{16} μs, an “under flow” is reported as (0000).

The ULI Force Probe takes several minutes to stabilize from the time power is first applied. The Hall effect sensor in it is temperature sensitive and has a small “self-heating” effect. The ULI Force Probe must be “initialized” before using this mode by selecting the proper input with the [S] command and then issuing the “P400<CR>” command.

Example:

In this example, a ULI force probe is attached to Port 1 and the Motion Detector is attached to Port 2. This is verified by checking the ID voltages with the [A] command. The sample period is set to 0.1 second. Before collecting data, the force probe is zeroed using the [P] command. The Motion Detector is aimed at a wall approximately 1.6 meters away and mode 9 is started. The same example is then repeated but with the values displayed in decimal format.

```
ULI2 Rev. 1.00
H4/3>a
027601F5
H4/3>efa
H4/3>t000190
H4/3>p400
Check
H4/3>m9
Data:
24110005
231E0005
22C90005

H4/3>d
D4/3>p400
Check
D4/3>m9
Data:
8924,5
8929,5
8897,5
8891,5
8914,5

D4/3>
```

Mode A - Read All Analog Inputs

Description:

This mode monitors all analog voltage inputs and one reference voltage with user selected sampling. The ULI$_{II}$ returns two additional reference voltages; V ref- and V ref+. The V ref value reported equals
(((V ref+) - (V ref-)) / 2).

Reporting Format: The count representing the voltage for each channel is displayed. The original ULI displays twelve counts while the ULI$_{II}$ displays 14 and in the following order:

A-to-D converter channel number	0	1	2	3	4	5	6	7	8	9	10	Vref	Vref -	Vref +
order in mode A list	1	2	3	4	5	6	7	8	9	10	11	12	13	14

connection on ULI$_{II}$		DIN4-ID	DIN4-Shield	DIN4-Vin	DIN3-ID	DIN3-Shield	DIN3-Vin	DIN2-ID Port2-ID	DIN2-Vin Port2-Vin	DIN1-Vin Port1-Vin	DIN1-ID Port1-ID	*	*	*
connection on original ULI					AUX-Vin		AUX-ID	DIN2-ID Port2-ID	DIN2-Vin Port2-Vin	DIN1-Vin Port1-Vin	DIN1-ID Port1-ID	*		
Analog Port pin #	2	14	3	13	4	12	5	11	6	10	7	16		

The *'s in the chart indicate internal reference voltages that are measured by the ULI. There is one measured reference voltage on the original ULI and three on the ULI$_{II}$:

Vref is a measurement of a test voltage produced by the analog-to-digital converter itself. This number should always be half the maximum Analog-to-Digital converter value ±1.2%. For the original ULI, this value is 512 ±12. Vref for the ULI$_{II}$ is 2048 ±4. **Vref-** is the low reference voltage or the lowest voltage measured by the Analog-to-Digital converter which is 0. Similarly, **Vref+** is the high reference voltage or the highest voltage measured by the Analog-to-Digital converter which, for the ULI$_{II}$ is 4095 and for the original ULI is 1023.

Ports Used:

All analog channels are read. All eleven channels can be accessed through the Analog Port on the original ULI (see *ULI Connectors* section for location and pin outs).

The ULI$_{II}$ does not have an Analog Port that allows access to all eleven channels. Ten of the eleven can be accessed and only through the DIN connectors.

Setup:

The setup required for this collection mode is to set the sample period and to set the display mode and precision. The I/O port to be used and how many bytes are used to report the values must be selected.

Associated Commands:

- Use the [B], [D] and [H] registers to set the display mode.
- Use the [C] command to set the number of bytes used for reporting voltages
  [C]=1 sets the maximum reported value as 255
  [C]>1 sets the maximum reported value as 1023 on the original ULI and 4095 on the ULI$_{II}$
- Use the [T] and [E] registers to set the sample period.
- Mode A is able to take advantage of Flash ([*] control) and Burst ([!] control) modes for extra fast data collection.

Exception Handling:

Not Applicable

Display Options:

Data is in a two-byte, 10-bit format or 12-bit on the ULI$_{II}$ if the [C] register is set > 1, or in one-byte, 8-bit format if [C]=1. The 8-bit data format is useful when you are trying to obtain the very fastest sampling possible. It has a resolution of 1 part in 256 or 20 mV.

Output:

Data may be reported in hexadecimal or decimal format. 12 values are reported by an original ULI. 14 values are reported by a ULI$_{II}$. The chart at the beginning of this section indicates the order in which the voltages are listed. Sampling skew (time difference between sampling ports) is approximately 1 μs.

Example:

In this example done with the original ULI, the time between samples is one second and the values are displayed in decimal format.

```
ULI Rev. 5.20
H4/3>t000f42
H4/3>d
2C01
D4/3>ma
Data:
1002,458,311,219,185,138,133,500,95,5,578,511
1002,451,301,207,170,124,121,500,94,5,578,511
1002,451,301,207,166,122,118,500,94,5,578,511
1002,451,301,207,166,122,118,500,94,5,578,511
1002,451,301,207,165,122,117,500,94,5,579,511
1002,451,301,207,165,122,117,500,93,5,578,511

D4/3>
```

Mode B- Resistance and Analog Input Reading

This mode is applicable only to the original ULI.

Description:

Mode B is a combination of Mode 3 and Mode 8 in that it will read both resistive inputs and both analog inputs, either from the DIN connectors or the RCA jacks. All 4 readings are reported each sample period even if one or both of the resistive inputs has infinite resistance (no connection). If the resistance is too high “FFFFh” is reported for that resistive input port.

Ports Used:

Resistive inputs and Port 1/DIN1 and Port 2/DIN2

Setup:

The only setup required for this collection mode is to set the sample period and to set the display mode and precision.

The resistance to be measured should be connected between +5V and the resistive input line. The outside of the resistive input RCA connectors is +5V.

Associated Commands:

- Use the [B], [D] and [H] registers to set the display mode.
- The [E] and [T] registers are used to determine the sampling interval. The [T] register also determines the time period after which the mode will give up waiting for the capacitor to charge and report an overflow.
- The [S] and [C] registers are ignored in this mode.

Exception Handling:

Not Applicable

Display Options:

Data may be reported in hexadecimal, binary, or decimal format.

Output:

The data are reported in this order: resistive port 1, resistive port 2, voltage port 1 and voltage port 2. The resistive input readings are reported as times in microseconds. If the resistance is larger than this mode can measure with the current [T] register setting, FFFFh will be reported. The voltages are reported as counts <0:1024> or <0:3FFh>.

Using decimal display, the count from both resistive inputs is displayed separated by a delimiter. Using the hexadecimal display, the data are displayed with 4 characters for each of the two inputs.

In the worst case, the resolution in the time reported for the capacitor to charge is about 10 µs Notes:

Example:

In this example, an original ULI reads the resistive and analog inputs every half second and display the values in decimal format. A 10 k variable resistor is connected to resistive input #1 and a 10 k, 10% carbon film resistor is attached to resistive input #2. A variable voltage source is connected to DIN1 and a constant, 1.23 volt source is attached to DIN2. During the example, the knob on the variable resistor is turned to different positions and the voltage at DIN1 is varied.

```
ULI Rev. 5.20
H4/3>efa
H4/3>t0007d0
H4/3>d
2C01
D4/3>mb
Data:
73,132,403,751
73,132,55,751
1,138,260,749
89,156,396,749
128,132,68,751
86,156,65,751
106,132,434,751
127,132,384,751
106,132,87,751

D4/3>
```

Mode C - Read Analog Inputs and Digital Input/Output

Description:

This mode was designed to monitor all digital lines (input and output) and the analog inputs of DIN1/port1 and DIN2/port2. This mode is useful for controlling experiments by setting up a feed back loop. For example, if the temperature inside a terrarium gets too cold (as measured with a temperature probe), digital output 1 could turn on a heater.

A “status byte” is used to show the state of the 2 digital inputs and 4 digital outputs. The status byte is assigned as follows:

Bit 7 - used for internal diagnostics
Bit 6 - undefined
Bit 5 - Digital Input #2
Bit 4 - Digital Input #1
Bit 3 - Digital Out #2
Bit 2 - Digital Out #1
Bit 1 - AUX Out #2
Bit 0 - AUX Out #1

The digital-out lines are controlled using escape sequences.

Ports Used:

Digital lines, Port 1/DIN1 and Port 2/DIN2

The Modular Phone Connectors (Ports 1 and 2) have digital input lines. If you have a probe plugged into one of the ports, it will affect the behavior of the matching RCA jack[6] digital input. Port 1 will affect Digital Input 1 and Port 2 will affect Digital Input 2. When using the Digital Out lines on the original ULI, remember that the outside connector on the Digital Out RCA jack is at +5 volts. If you connect an LED/resistor combination across this jack, the LED will light when the Digital Out line goes low.

[6] This applies to the original ULI only.

Setup:

The setup required for this collection mode is to set the sample period and to set the display mode and precision. The I/O port to be used and how many bytes are used to report the values must be selected.

Associated Commands:

- Use the [B], [D] and [H] registers to set the display mode.
- Use the [E] and [T] registers to determine the sampling interval. The [T] register also determines the time period after which the mode will give up waiting for the capacitor to charge and report an overflow.
- Use the [C] command to set the number of bytes used for counting. If too few internal accumulators are used to display the time, zero will be returned.
- The [S] register is used to select the analog input. Analog inputs 1 and 2 are allowed. The most useful [S] register settings are identical to Mode 8 ([S] = 1 for Port 1 only, [S] = 2 for Port 2 only, [S] = 3 for both Port 1 and Port 2).

Exception Handling:

Not Applicable

Display Options:

Data may be reported in hexadecimal, binary, or decimal format.

Output:

Data may be reported in hexadecimal or decimal format. Sampling skew (time difference between sampling ports) is approximately 1 s.

The format of the output is: Status Byte, Port 1, Port 2

Example:

The following example, the voltage on DIN1 only is read along with the digital lines. Notice, the digital inputs do not change.

```
D4/3>s1
D4/1>mc
Data:
48,1002
48,433
48,129
48,200
48,403

D4/1>
```

The following example, the voltage on DIN1 and DIN2 are read along with the digital lines. Again, the digital inputs do not change.

```
D4/1>s3
D4/3>mc
Data:
48,511,229
48,170,77
48,126,58
```

In this example, the voltage on DIN1 and DIN2 are read along with the digital lines. Notice, the output is now in hexadecimal format.

```
D4/3>h
H4/3>mc
Data:
3000210012
3000500025
30008B0040
3000B70053
```

```
30019000B3
3001AF00C0
3000B80053

H4/3>
```

Mode D - Sequential Pulse Period Timer

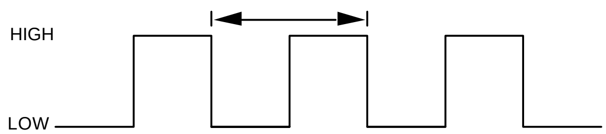
Description:

This mode was designed to time sequential pulses much like Mode 4 but is used when no dead time can be tolerated between successive time interval measurements. It was originally developed to study the correlation between the duration of sequential events in chaotic systems. This mode can be used with a ten spoke pulley in an Atwood's machine to measure acceleration due to gravity.

This mode uses the Digital Input 1 and reports the period of the applied signal. The time measured is from falling edge to falling edge. Times are stored in the buffer and are reported all at once when the buffer is full. Two timers are used so that pulses are never missed.

After the buffer is filled, data collection is stopped. The buffer is emptied and then data collection continues.

<Ctrl>+C ends data collection without reporting any data currently in the buffer.



Timing using Digital Input #1

Ports Used:

Digital Port 1

Setup:

The setup required for this collection mode is to set the sample period and to set the display mode and precision. You must also set the Digital Input line to be used. Exception handling may also be set.

Associated Commands:

- Use the [B], [D] and [H] registers to set the display mode.
- Use the [C] command to set the number of bytes used for counting. If too few internal accumulators are used to display the time, zero will be returned.
- The buffer size is determined by the [R] register. In most cases (unless the [C] register is set to something other than 4) each page assigned to the buffer will allow storage of 64 times.
- The [J] register (error status) is used to test when the buffer is full.

Exception Handling:

If you are in Hexadecimal or decimal mode and the [J] register is set to trap "Timer 1 rollover," it will be reported as an "Err8" after the buffer is full.

Display Options:

Data may be reported in hexadecimal, binary or decimal format. Note that nothing will be displayed until the buffer is full and then all the times will be displayed.

Output:

The times collected in this mode have a resolution of about 1 µs. The data are displayed as a time in microseconds. Using hexadecimal display, the number of digits displayed depend on the display mode and the value of [C].

Example:

This example demonstrates how to use this mode to monitor a ten spoke pulley on an Atwood's machine. The buffer is set to store 2 pages of four byte times. This is sufficient to measure 128 cycles or 12.8 revolutions of the pulley. (Not all of the data are presented here.)

```
ULI2 Rev. 1.00
H4/3>md
Data:

H4/3>r02
H4/3>md
Data:
0000442C
00004456
000044D3
0000456B
0000C016
0000C5C8
0000CC85
0000D575
0000DCD6
0000EA42
0000F4F8
00010537
00011411

H4/3>
```

Mode E - Rotary Motion

[Original ULI ROM 5.12 or later required (5.2+ suggested) or ULI$_{II}$]

Description:

This mode is designed to count the events generated by an optical encoder in a Rotary Motion Probe connected to Port 1. Mode F is more flexible, in that it can be used with a Rotary Motion Probe connected to either Port 1 or Port 2, and it can collect analog input data at the same time. You may want to use Mode F instead of Mode E.

It expects to receive pulses on two inputs of Port 1. The ID input line is used to signal a one count change in position in the “positive” or clockwise direction and the Digital Input line signals a change in the “negative” or CCW direction. The ULI keeps track of the net change in position over a time period and reports this number at the user specified intervals.

The resolution of the Rotary Motion Probe is controlled by the Digital Out line. It is usually is 0 but can be set to 1 to increase the resolution of the data by 4 times.

The ULI uses a hardware counter in the “negative” or CCW direction and has a top speed of over 100,000 counts per second.

The ULI uses a software counter in the “positive” or clockwise (using our hardware) direction which results in a maximum measurable rate of about 15,000 events per second. Once that rate is exceeded, the ULI may begin to miss events and the net position change will no longer be correct.

The number returned by the ULI is the net change in position over the time period. It is up to the user to track the net change since the start of data collection.

Sometimes the first data record will report a small number other than 0. Ignore this number. It is the result of hardware setup. Treat the first record as 0 and measure you net position changes starting from that point.

Another way to measure the same type of data is with Mode 1. That mode allows very accurate timing of changes in state of the digital input lines. If you are interested in slow rotations, Mode 1 may give more accurate change of state data than Mode E. This is much better if you are interest in the velocity or acceleration of rotation (how fast is the pulley turning instead of just the position). You can get fairly good velocity and acceleration data for rapid rotations but for slow rotations the rough granularity of the counts/time period data yields “noisy” derivatives.

Ports Used:

Port 1

Setup:

The only setup required for this collection mode is to set the sample period and to set the display mode. You may also want to set the resolution of the optical encoder.

Associated Commands:

When using this mode, leave the [S] register set to 3 and the [C] register set to 4. You can use binary mode for data collection with this mode but error trapping is much easier in the hexadecimal or decimal display modes. Since Mode E generally doesn’t generate too much data, this is recommended.

Most of the rotary encoder probes have two resolutions. By using the Escape codes, the digital line can be used to select between two of these resolutions. Setting the digital out line HIGH selects a resolution with 4x the counts per rotation on the PASCO or Tufts Rotary Motion probe, but limits the maximum speed.

Here are the commands use to set the Rotary Motion Probe resolution. Note that these commands are case sensitive. They can be issued at any time, even in the middle of data collection.

<ESC>A	sets digital output 1 high and high resolution
<ESC>B	sets digital output 1 low and low resolution

Exception Handling:

If the number of events occur too quickly (for example, more than 15,000/s in positive direction), the ULI will start to miss events. This type of error can be trapped by setting the **EI1 Late / Timer 1 Late Read** bit of the [J] register before data collection. This trap will terminate data collection with a “Err08<CR><LF>”. Catching the error message is much easier in ASCII modes (hexadecimal or decimal) and we suggest you work in these modes first and only switch to binary if needed.

After data collection has ended with an Err condition, the ULI should be reset with Mode 0.

Display Options:

Data may be reported in hexadecimal, binary or decimal format.

Output:

The net position change from previous reading is reported. In hexadecimal mode, the data are formatted as xxxxxxxx, where xxxxxxxx = net position change from previous reading

In decimal mode the net position change from the previous reading is simply reported.

As you use this mode, you will probably want to keep a running total of counts so you know the exact position of the Rotary Motion Probe.

Example:

In this example, the time interval is set to 1 reading/second. The shaft of a Rotary Motion probe is rotated slightly. Comments are placed next to the collected data to indicate how data should be interpreted.

```
ULI2 Rev. 1.00
H4/3>efa
H4/3>t000fa0
H4/3>me
Data:                 Position Change    Current Position
00000000                    0000               0000
00000005                   +0005              +0005
00000009                   +0009              +000E
```

```
00000004                +0004           +0012
FFFFFFF7                -0009           +0009
FFFFFFFB                -0006           +0003
FFFFFFF9                -0007           -0004
00000000                 0000           -0004

  H4/3>
```

Mode F - Rotary Motion and Analog Input

[Original ULI ROM 5.12 or later required (5.2+ suggested) or ULI_{II}]

Description:

This mode is designed to count the events generated by an optical encoder connected to Port 2 and to collect analog data from Port 1/DIN 1. This mode is a hybrid of Mode 8 and Mode E so that an analog voltage producing probe can be used with a Rotary Motion probe in physics experiments. Even though Port 2 is normally used for the Rotary Motion Probe, Port 1 can be used for it .

Mode F expects to receive pulses on two inputs of the port assigned to the Rotary Motion Probe. The ID input line is used to signal a one count change in position in the “position” or clockwise direction and the Digital Input line signals a change in the “negative” or CCW direction. The ULI keeps track of the net change in position over a time period and reports this number along with an analog input from Port 1/DIN 1 at the user specified intervals.

The resolution of the Rotary Motion Probe is controlled by the Digital Out line of its port. This line is usually is set to 0 but can be set to 1 to increase the resolution of the data by 4 times.

The ULI uses a software counter in the “positive” or clockwise (using our hardware) direction which results in a maximum measurable rate of about 15,000 events per second. Once that rate is exceeded the ULI may begin to miss events and the net position change will no longer be correct.

The ULI uses a hardware counter in the “negative” or CCW direction and has a top speed of over 100,000 counts per second.

Ports Used:

Normally: Port 1/DIN1 (for analog input) and Port 2 (for Rotary Motion).

Port 1 can be used for the Rotary Motion Probe.

Setup:

The only setup required for this collection mode is to set the sample period, the display mode, and the port of the Rotary Motion Probe. You may also want to set the resolution of the optical encoder.

Associated Commands:

When using this mode, leave the [S] register set to 3 and the [C] register set to 4. You can use binary mode to data collection with this mode but error trapping is much easier in hexadecimal or decimal mode. Since Mode F generally doesn’t generate too much data, this is recommended.

Most of the rotary encoder probes have two resolutions. By using the Escape codes, the digital line can be used to select between two of these resolutions. Setting the digital out line HIGH selects a resolution with 4x the counts per rotation on the PASCO or Tufts Rotary Motion probe, but limits the maximum speed.

Here are the commands use to set the Rotary Motion Probe resolution. Note that these commands are case sensitive. They can be issued at any time, even in the middle of data collection.

If using Port 1 for Rotary Motion Probe::
<ESC>A sets digital output 1 high and high resolution
<ESC>B sets digital output 1 low and low resolution
If using Port 2 for Rotary Motion Probe:
<ESC>C sets digital output 2 high and high resolution
<ESC>D sets digital output 2 low and low resolution

Exception Handling:

If the number of events occur too quickly (>15,000), the ULI will start to miss events. This type of error can be trapped by setting the **EI1 Late / Timer 1 Late Read** bit of the [J] register before data collection. This trap will terminate data collection with a “Err08<CR><LF>”. Catching the error message is much easier in ASCII modes (hexadecimal or decimal) and we suggest you work in that mode first and only switch to binary if needed.

After data collection has ended with an Err condition, the ULI should be reset with Mode 0.

Display Options:

Data may be reported in hexadecimal, binary or decimal format.

Output:

The net position change from previous reading is reported first, followed by the analog input reading. In hexadecimal mode, the data are formatted as xxxxxxxxyyyy

where: xxxxxxxx = net position change from previous reading
yyyy = analog input (10 bit on ULI, 12 bit on ULI_{II}).

In decimal mode the net position change from the previous reading is simply reported, then the delimiter, then the analog reading.

As you use this mode, you will probably want to keep a running total of counts so you know the exact position of the Rotary Motion Probe.

Example:

In this example, the shaft of a variable resistor is attached to the shaft of a Rotary Motion Probe (in Port 2) and the angle is varied. The sample rate is set to 1 sample/second. Comments are placed next to the collected data to indicate how data should be manipulated.

```
ULI2 Rev. 1.00
S2
H4/2>efa
H4/2>t000fa0
H4/2>mf
Data:              Position Change    Current Position   Analog
0000000000FF             0000                0000          00FF
0000000503B6            +0005               +0005          03B6
000000090225            +0009               +000E          0225
00000004014E            +0004               +0012          014E
FFFFFFF70115            -0009               +0009          0115
FFFFFFFB00B3            -0006               +0003          00B3
FFFFFFF900B3            -0007               -0004          00B3
0000000000B8             0000               -0004          00B8

   H4/3>
```

Mode FF - Test ULI

Description:

This is a diagnostic function used during ULI testing. Once invoked, the status LED will cycle through a binary count pattern at the rate of one count every 1.049 seconds (2^{20} μs). Since most of the ULI's resources are used in performing this operation, it is a good indicator of the “wellness” of the unit. The LED’s allow the operator to monitor the operation of several ULI's at the same time without requiring that a terminal be connected to each one. (A terminal is required only to start up and shut down the units but, they may be disconnected while the test is being run.).

Register Commands

The ULI uses internal programs for all its modes. As in other computer programs, variables are used. In the ULI, the variables are called *registers* that determine how these programs run. These registers specify everything from how often to take samples to how the output is presented. The ULI registers can be read and written using the same basic command.

Register C - Data Word Length

The [C] register sets the size of internal data accumulators used by the ULI. These data accumulators can be 1 to 4 bytes. If the accumulator is set to 1 byte wide it can hold numbers from 0 to 255 (FFh). It will overflow if it counts above FFh. If the [C] register is set for 4, the accumulator is 4 bytes wide and it can count up to FFFFFFFFh (2^{32})before overflowing. The contents of the [C] register is always reported in the ULI prompt (character 2). The [C] register is set to 4 when the ULI is first turned.

Write:

To set the value of the [C] register, type

Ca <CR>

where: a = the size of internal data accumulators, <1 : 4>.

Example:

In this example, the number of bytes used for the counter/timer will be set to 4 and will have the range of 0 to FFFFFFFFh. The timer/counter will overflow at 2^{32} (4294967296). The [C] register is then set to 1 and will have a range of 0 to FFh. The timer/counter will now overflow at 2^{8} (256).

Mode 1 data collection will be used to make the point (see the description of Mode 1 in the *Data Collection Modes* section for more details). The number before the last two characters demonstrates the overflow feature.

```
ULI Rev. 5.20
H4/3>m1
Data:
0000001000
0016AC9F01
0016E4FE00
0017286F01
00175F6900
00180C1201

H4/3>c1
H1/3>m1
Data:
2000
C501
7800
9D01
ED00

H1/3>
```

Register E - Time Base

The [E] register sets the time base for analog data collection. When the ULI is turned on, the [E] register is set at 0 which is treated as 100h (256) and specifies 256 micro-seconds. The time between readings for many data collection modes is specified by the value of the [T] register multiplied by the time base specified by the [E] register.

The [E] register should be used only to fine tune the clock rate and not to speed it up. Settings less than 128 (80h) should be avoided.

Read:

The command to read the [E] register is

`E<CR>.`

The ULI will return the following:

aa<CR><LF>

where: aa = the value of the [E] register.

Write:

To set the value of the [E] register, type

Eaa <CR>

where: aa = the time base, <00 : FFh>.

When setting the time base, a 2-digit hexadecimal number must be entered. An error will result if any of the characters is a space. If any of the characters are outside the range for hexadecimal characters, it will be ignored.

Example:

This example demonstrates how the time base can be changed to 250 microseconds by assigning the [E] register the value Fah. This is sometimes used to make the time intervals come out as more precise. The error trapping feature is also demonstrated when, first, only one hexadecimal value is entered and, then, an invalid hexadecimal value is entered.

```
...
H4/3>EFA
FA
H4/3>E4
Error                      only one character entered
H4/3>EZ4                   the Z is ignored
H4/3>E
04
H4/3>EFZ                   the Z is ignored
H4/3>E
0F
H4/3>...
```

Register I - Digital and Analog Triggering Conditions

This command is complex. You may need to read this section a couple of times in order to really grasp the concepts involved. Also, you should look over the section on the [V] register.

You may want to select a trigger for data collection in some modes. Triggering works with all buffering modes, including flash and burst modes. If you set one or more trigger conditions you will see the “Data:” message, and then the ULI will pause until the trigger event occurs or a <Ctrl>+C is received. Once sampling has been started, it will continue at the rate determined by the [T] and [E] registers in the normal buffer mode or by the [E] register in the flash or burst mode. The “Status” light on the ULI will turn on when the trigger conditions have been met, and turn off at the completion of the upload.

In flash and burst mode with a trigger set, the first 256 bytes of data collected will be pre-trigger data. This data will look the same as the post-trigger data. If the trigger occurs before the pre-trigger buffer could be filled, the unused leading bytes will be “0”s so you should ignore any leading zeroes in pre-trigger data.

WARNING: Pre-trigger data are collected at a maximum rate of about 6000 points/second. You may use the [J] register to make sure your pre-trigger data are being collected as fast as your post trigger data.

There are two types of trigger events:

1. Analog triggering when the input is either greater than a high trigger or is less than a low trigger.
2. Pulse triggering on the digital inputs.

Every trigger has a separate enable bit. The trigger is enabled or disabled by setting or clearing the appropriate bit of the [I] register.

a7	a6	a5	a4	a3	a2	a1	a0

where: a0 = Digital Input 1 Transition (1 = positive, 0 = negative)
a1 = Digital Input 1 Enable (1 = enabled, 0 = disabled)
a2 = Digital Input 2 Transition (0 = negative)

a3 = Digital Input 2 Enable (1 = enabled, 0 = disabled)

a4 = Analog Voltage 1 Low Trigger
a5 = Analog Voltage 1 High Trigger
a6 = Analog Voltage 2 Low Trigger
a7 = Analog Voltage 2 High Trigger

The digital input trigger states will **not** work correctly if digital input #2 is set to trigger on a positive transition.

When triggering on an analog level, you must set the level with the [V] register.

The analog triggers may be set up to be either single-stage triggers with up to two conditions (greater than voltage A or less than voltage B) or a two-stage trigger with each stage having a single condition.

Two-stage triggers can work as a “window” trigger where any voltage between the two conditions will trigger data collection. The two-stage trigger can also be used in the “slope” mode where the slope is set by **b6** or **b4**. If a positive slope is chosen, the ULI waits until the analog input value is less than the lower of the two trigger voltages for that port. Once the voltage is less than the trigger level, the ULI waits until the input level exceeds the larger of the two trigger voltages and then begins collecting data. The negative two-stage trigger is just the opposite.

If you want to set up a two-stage trigger, a second byte needs to be sent to the [I] register. The second byte has the format

b7	b6	b5	b4	b3	b2	b1	b0

where: b0 = trigger - Port 1 (set by ULI)
b1 = pretrigger received - Port 1 (set by ULI)
b2 = same as b0 for Port 2
b3 = same as b1 for Port 2

These four bits should always be set to 0. They serve to return information about the trigger status after data collection (like the second byte of the I register query command).

b4 = 1 = positive slope - Port 1
= 0 = negative slope - Port 1
b5 = 1 = slope mode - Port 1 (ignore analog settings in byte 1)
= 0 = window mode - Port 1 (refer to settings in byte 1)
b6 = 1 = positive slope - Port 2
= 0 = negative slope - Port 2
b7 = 1 = slope mode - Port 2 (ignore analog settings in byte 1)
= 0 = window mode - Port 2 (refer to settings in byte 1)

Setting the slope bits if they are not setting the enable bit does not enable slope mode triggering. The [I] register command can include a second byte if the two-stage (slope) analog triggers are being used. Don't send this byte unless you are using it, i.e., don't send I0000 to clear those triggers, send I00.

Read:

The command to read the [I] register is

I<CR>.

The ULI will return the following is a single-stage trigger is setup:

aabb<CR><LF>

where: aa = the enable data

bb = the first trigger level

Is a two-stage trigger is setup, the ULI will return the following:

aabbcc<CR><LF>

where: aa = the enable data

bb = the first trigger level

cc = the second trigger level

Write:

To set the value of the [I] register, type

Iaabb <CR>

where: aa = the basic triggering mask, <00 : FFh>,

bb = two-stage trigger mask.

The analog triggers look at only the TOP 8 bits of the data. Remember this when setting the trigger values. If you are going to collect 8-bit data, the trigger values are the same as the raw values you see. If you are collecting 10-bit data with the original ULI, then you must shift your desired trigger value right by two bits to get your trigger value. Similarly, if you are collecting 12-bit data with the ULI_{II}, then you must shift your desired trigger value right by four bits to get your trigger value.

Example:

Due to the complexity of this command, several examples of it's use are demonstrated here. All examples will be described as they would be used in data collection mode 8. The first example demonstrates how to trigger data collection with a negative transition on digital line 1. Recall that bit **a1** must be set to 1 to enable triggering with digital input 1 and that bit **a0** sets the transition. In this case, it is set to 0.

```
...
H4/3>i02
H4/3>m8
...
```

The next example demonstrates how to set up a trigger with a voltage above 2.56 volts on analog input 1. Bit **a5** must be set to 1 to enable triggering with analog input 1 high trigger and that voltage level is set with the [V] register. This corresponds to a 128 (80h).

```
...
H4/3>i20
H4/3>v00008000
H4/3>m8
...
```

This next example demonstrates how set up a two-stage, window trigger for DIN1/Port 1. The trigger will start data collection if the input voltage on DIN1/Port1 is between an upper and a lower voltage. In this example, the lower voltage will be 2.30 volts and the upper voltage limit is set to 2.50 volts. The corresponding trigger values placed in the [V] register are 115 (73h) and 125 (7Dh), respectively. To setup the trigger, both bits **a4** and **a5** must be set to 1 with all other **a** bits set to 0, and bit **b5** set to 1 with all other **b** bits set to 0. Referring to the description of the [V] register, the upper and lower trigger values need to be set in the [V] register for Port 1. This means four bytes need to be written to the [V] register.

```
...
H4/3>i3010
H4/3>v00007D73
H4/3>m8
...
```

To set up the same two-stage, window trigger for DIN2/Port2, the following commands are used.

```
...
H4/3>iC080
H4/3>v7D730000
H4/3>m8
...
```

Setting up a two-stage, slope trigger is very similar to the window trigger in that two voltage limits are setup. The only difference is that the signal must pass through both voltage limits in proper sequence. For example, a positive slope trigger is setup on DIN1/Port1. This means the signal must pass through the lower limit first then through the upper limit.

```
...
H4/3>i0030
H4/3>v00007D73
H4/3>m8
...
```

These triggering modes can be compounded to create a very complex triggering system. This type of trigger takes a long time to evaluate and can lead to slowed pretrigger data collection. Remember that all of these conditions must be met at the same time in order for data collection to begin. An example of a compound trigger is a negative transition on digital input 1, a positive slope trigger on DIN1/Port1 and a window trigger on DIN2/Port2.

```
...
H4/3>iC230
H4/3>v7D737D73
H4/3>m8
...
```

Register J - Error Trapping Mask

Many modes, modes 2, 8, and 9 in particular, have powerful error trapping features. The [J] register is devoted to error flags and every error has a separate enable. An error flag is enabled or disabled by setting or clearing the appropriate bit of the [J] register. The ULI has 8 error flags which are collected into one byte known as the error mask. If any bit in the error mask is set and that error occurs during data collection, the data collection is immediately aborted. Not all error conditions relate to each collection mode.

The following is a bit-by-bit description of the error flags and the conditions that will generate them. This is the general case and represents what happens with most modes. Exceptions, if any, will be noted in the individual mode documentation.

j7	j6	j5	j4	j3	j2	j1	j0

where:

j0 **State of digital input A -** Used internally for phasing of an up/down counter. Normally, these bits should be ignored and error trapping should be used only for diagnostic purposes.

j1 **State of digital input B -** Used internally for phasing of an up/down counter. Normally, these bits should be ignored and error trapping should be used only for diagnostic purposes.

j2 **EI0 Late / Timer 0 late read -** Timer 0 or External Interrupt #0 was serviced late. Usually occurs when the event being monitored happens too fast for the program to keep up. Where possible, reduce the program overhead by using binary reporting format or slowing down the experiment.

j3 **EI1 Late / Timer 1 late read -** Timer 1 or External Interrupt #1 was serviced late. Usually occurs when the event being monitored happens too fast for the program to keep up. Where possible, reduce the program overhead by using smaller data frames, binary reporting format, or slowing down the experiment.

j4 **Timer 0 rollover -** Timer #0 has overflowed. Readings may be compromised. Usually occurs when the event being monitored is too slow for the time base being used. Where possible, use a longer time base.

j5 **Timer 1 rollover -** Timer #1 has overflowed. Readings may be compromised. Usually occurs when the event being monitored is too slow for the time base being used.

j6 **Transmit Overflow -** This bit is set when a collection mode attempts to add additional data when the output buffer is full. When this occurs, the program will loop until enough additional buffer space becomes available to store the current reading. Once this is done, the error trap condition is tested and, if the matching bit was set in the mask register, the program sets an internal “termination pending” flag, shuts down the mode currently in operation, and enters an idle loop, waiting for the balance of output buffer data to be shifted out before returning to the command prompt. This procedure is followed to insure that no valid data are lost and providing the option of automatically terminating the program when data integrity can no longer be insured.

j7 **Data Frame Error -** A data frame is the data field width and is generally measured in bytes. This bit is set when a measurement results in more data than will fit in the specified data frame. For example, in Mode 2, the data frame is a fixed 2 byte format. If more than 0FFFFh s elapses (while waiting for the echoed pulse), bit **j7** is set.
In some modes, the size of the data frame can be set with the [C] command. In others, where the nature of the measurement limits the valid range of data (such as Mode [C]) it is fixed.
Note: If the data frame is set to 4 bytes (C=4), data frame errors are typically not flagged (exceptions are noted where applicable).

Read:

The command to read the [J] register is
`J<CR>.`

The ULI will return the following:

aabb<CR><LF>

where: aa = the current error mask.
bb = the actual error flags

Write:

To set the value of the [J] register, type

Jaa <CR>

where: aa = the hexadecimal representation of the error mask, <00 : FFh>.

Writing to this register sets the error mask. To put it another way, this register tells the ULI which error conditions to look for. When setting the error mask, a 2-digit hexadecimal number must be entered. An error will result if either character is a space. If either of the characters are outside the range for hexadecimal characters, it will be ignored.

Example:

The most frequently used [J] command is J40<CR>. This command sets the [J] register to 40h or 01000000. This sets just the **j6** bit high. The **j6** bit corresponds to transmit Overflow (buffer full). After the [J] register is set to 40h, data collection modes will abort whenever the buffer fills up.

In mode 8, **j2** is used to signal a late read in the trigger routine. An error condition will occur if the trigger conditions were too complex to be evaluated in the time allowed for data collection (and therefore the pre-trigger data are **not** collected at the same speed as the post trigger data). You can use this to find the fastest possible data rate for flash or burst transfer modes that allows a given set of trigger conditions. If you don’t care about the pre-trigger data, then this error won’t matter. The trigger will still be tested for all enabled conditions; it will just run a bit slower than the data collection once it starts. Also in mode 8, setting the **j3** bits will abort data collection when a late read is detected. It means that you are going too fast for the data frame (data size/number of ports combination) selected. Again, you can experiment with this to test for maximum speeds in burst mode.

```
ULI Rev. 5.20
H4/3>R20        ; set large ring buffer so buffer overflow isn't a  ;
problem
H4/3>S1         ; lets collect from port 1 only
H4/1>J08        ; trap data collection too slow
H4/1>T000010
H4/1>EFA        ; 16 * 250 = 4000us, 1000000/4000 = 250 pts/sec
H4/1>m8
Data:
004A
004A
004A
004A
[data will be collected without problem]
[use CTRL-C to stop]

H4/1>T000004    ; 4 * 250 = 1000us, 1000000/500 = 1000 pts/sec
H4/1>EFA        ; always reset the E register, some errors leave it "bad"
```

```
H4/1>m8
Data:
004A
Err08           ; ULI can't keep up with selected data rate

H4/1>J
0800            ; note that ULI only reports this error after data
                ; collection now that it has been reported once the
                ; [J] register is cleared
H4/1>B          ; less overhead in binary mode (no converting to
                ; ASCII chars)
B4/1>EFA        ; reset [E] register
B4/1>m8
Data:
}j]TNIDA><:7643211122223322100/...../0011110//.------../00000/..

[or something similar, but without an error being generated so it
will run until you stop it. The point being that binary mode has less
overhead than in hexadecimal mode and so (in this case) the ULI can keep
up.]
[use CTRL-C to stop]
```

Register K - Programmable Amplifier Settling Time

The [K] register is used to control the zeroing time when the [P] register is used. This register is used only when a ULI Force Probe is being implemented in an experiment. A settling time was added when it was found that the delays required to get some probes to “settle” after an adjustment to the offset varied between probes. It is good practice to set the zeroing factor with the smallest value of possible. In the current ULI software, the [K] register has a setting of 8 even it could be less “most of the time.”

Read:

The command to read the [K] register is

`K<CR>.`

The ULI will return the following:

aa<CR><LF>

where: aa = number of microseconds to wait for a probe to settle to an offset value.

The result of reading the [K] register will be reported in hexadecimal format if the display setting is set for either hexadecimal or decimal. The results of reading the [K] register will not be reported if you are in the binary display mode.

Write:

To set the value of the [K] register, type

Kaa <CR>

where: aa = the number of microseconds, <00 : FFh>.

When setting the amplifier, a 2-digit hexadecimal number must be entered. An error will result if either character is a space. If either of the characters is outside the range for hexadecimal characters, it will be ignored.

A value of 00h sets the longest setting time of 256 microseconds. A value of 01h sets the fastest setting time and will not correctly zero any model of force probe.

All force probes will settle with a [K] register value of 09h. The PA-2 force probes generally require a 6 s settling time while FP-IIs and PA-I style force probes generally require a settling time of 8 s. A value of 0Fh is sometimes used as a second (MUCH slower) attempt for FP-II and PA-I/II force probes.

Example:

To see the various speeds, try this with a FP-II in port 1 (if you forget the FP-II, you should get a CHECK message from ULI after P000). In this example, the ULI will be configured to select port 1 only, set the settling time for the smallest delay and then zero the FP-II on port 1. The settling time will then be changed to show how a difference in settling time effects the final A/D “zero” reading.

```
ULI Rev. 5.20
H4/3>s1
H4/1>k1
09
H4/1>p000
8230000205

H4/1>k08
H4/1>p000
8230000202
H4/1>
```

Register P - Programmable Amplifier Gain and Offset Settings

This command allows you to set the gain and offset of a programmable amplifier (PA) if one is attached to your ULI. This register is used only when a ULI Force Probe is being implemented in an experiment. The [S] register specifies which port is to be used. If [S] = 1 or 0 then the PA is in port 1. If [S] is set to 2 or 3 then the PA is assumed to be in port 2.

A PA will work in either Port 1 or Port 2. The PA may be plugged in or unplugged while the power is on.

You **must** have a ULI_{II} or original ULI with ROM version 4.08F or later for the Programmable Amplifier to work correctly.

By setting the gain to twice the desired range, the resolution will be cut in half. This guarantees the entire range of the PA will be measured even if it was zeroed any place inside the range.

Make your software smart enough to automatically adjust the offset so that it is always set to the center of your desired range.

The exact relationship between the “gain number” and the real gain is unknown but, as one goes up the other goes up.

Read:

The command to read the [P] register is

`P<CR>.`

If you do not have a PA, this command has no function and returns “Error”. If a PA is found, after a few seconds the ULI will return the following:

aaabbbcccc<CR><LF>

where: aaa = the offset setting that “zeroed” the PA-1; <000h : FFFh>.

bbb = the “Gain” used; <000h : FFFh>.

cccc = the final A/D “zero” reading. It should be near 0200h (512 decimal).

Write:

To set the value of the [P] register, type

Paaa <CR>

where: aaa = the PA gain, <000 : FFFh>.

When setting the gain, a 3-digit hexadecimal number must be entered. An error will result if any of the characters is a space. If any of the characters is outside the range for hexadecimal characters, it will be ignored. After the gain is set by the ULI, it will return the following:

aaabbb<CR><LF>

where: aaa = the offset setting, <000h : FFFh>
bbb = the gain setting, <000h : FFFh>.

These values should be noted so that the probe's offset/gain settings can be specified in future uses.

To set the PA to a known state (say when loading a calibration or just before collecting data), type:

Paaabbb <CR>

where: aaa = the offset setting, <000h : FFFh>
bbb = the gain setting, <000h : FFFh>.

We recommend that when setting the PA to a known state, it directly precedes the data collection mode command. This is because some commands, such as the [A] register, disrupt the PA 's internal settings. If you use the [A] register, operation of the PA will be disrupted until it is re-set.

Always zero the PA in the middle of its range. If you are able to save and load all the information needed to initialize the PA (both the gain and offset), you can calibrate ahead of time with the PA zeroed and have the user load the calibration from disk.

Example:

In this first example, the Gain Setting of the PA attached to Port 1 will be set to 320.

```
H4/3>S1
H4/3>P320
H4/3>
```

In this example, data will be collected with a Force Probe in Port 1 and a motion detector in Port 2. Before data are collected with the Force Probe, it needs to be zeroed and the gain set to 400h

```
ULI2 Rev. 1.00
H4/3>t000fa0
H4/3>efa
H4/3>s1
H4/1>p400
8604000822
H4/1>s3
H4/3>m9
Data:
14290806
142A07FC
14290801

H4/3>
```

Register R - Number of RAM Pages for Buffer

The ULI has 8KB RAM built in that may be used for a number of different purposes. Its default use is to hold outgoing data that the host computer is not yet ready to received and is configured as a ring buffer. You can turn this on and off with the "/" and "\" commands. This buffer defaults to a size of one page where each page is 256 bytes long.

Flash and Burst mode use the RAM differently. You **must** write to the [R] register to set aside pages of RAM for these modes to use. They will fill that buffer, stop and send the data to the computer. Burst mode will start collecting again. This buffer has two parts – an automatic, one-page (256 byte) pre-trigger buffer and a user-determined number of post-trigger pages.

The size of the data buffer can be set from 1 to 32 (20h), 256-byte pages.

Read:

The command to read the [R] register is

`R<CR>.`

The ULI will return the following:

aabbcc<CR><LF>

where: aa = the total RAM pages,
bb = the total pages used by user programs,
cc = current buffer pages. **cc** will always equal at least one for a pre-trigger buffer page.

The results of reading the [R] register are reported in hexadecimal format if the display setting is set for either hexadecimal or decimal. The results of reading the [R] register are not reported correctly if you are in the binary display mode.

Write:

To set the value of the [R] register, type

Raa <CR>

where: aa = number of RAM pages reserved as a communications buffer, <00h : 20h>.

When setting the time base, a 2-digit hexadecimal number must be entered. An error will result if any of the characters is a space. If any of the characters is outside the range for hexadecimal characters, it will be ignored.

The flash and burst modes use the first page to record pre-trigger data if any triggers are enabled.

Example:

When the ULI is first turned on, reading the [R] register returns 200001 indicating 20h pages of total RAM with zero devoted to user programs and one page set aside for the buffer.

To set aside all 8K for a burst mode buffer send “R20<CR>”.

This example specifies that 16 (10h) pages of the ULI’s RAM will be used for the data buffer.

```
ULI Rev. 5.20
H4/3>r10
H4/3>
```

Register S - Selects Active Ports

The [S] register selects the active I/O port(s). The first two bits of the [S] register represent the Analog input lines of the first two DIN connectors. Bit 0 represents Port 1 or DIN 1 and bit 1 represents Port 2 or DIN 2. On the original ULI, bit 1 also represents the AUX Port.

The most frequently used [S] register settings are:

1 only port 1 active
2 only port 2 active
3 both ports 1 and 2 active

The allowed range is <1 :3h>.

The value of the [S] register is always reported as character 4 of the prompt. When the ULI is first turned on, the [S] register is set to 3.

On some older ULI models, the [S] register can also be set to 0. This has the same effect as setting it to 3 (both ports 1 and 2 active).

Write:

To set the value of the [S] register, type

Sa <CR>

where: a = the active I/O port(s), <1h : 3h>.

When setting the time base, a 2-digit hexadecimal number must be entered. An error will result if any of the characters is a space. If any of the characters is outside the range for hexadecimal characters, it will be ignored.

Example:

In this example, the [S] register will be set to 1 so that only I/O port 1is active. Notice how the prompt changes after this command is sent.

```
ULI Rev. 5.20
H4/3>s1
H4/1>
```

Register T - Set Timer Period

The [T] register controls the timer period. In most of the data collection modes, this sets the time between sensor readings. The [T] register can be set to any value in the range <000000h : FFFFFFh>. This register is multiplied by a fixed time increment specified by the value in the [E] register (usually set to 256 µs or 250 µs) to determine a time between readings. Increasing the [T] value slows down data collection. The details of how the [T] register controls the timing of data collection vary with the data collection mode.

When the ULI is first turned on, the [T] register is set to 000000h which is treated as 1000000h. This leads to the longest possible sample period of over an hour.

[T]=000001h is the minimum time base (fastest sampling) and increasing [T] slows sampling down. The minimum value of [T] is determined by the average rate at which the host computer can absorb the data stream being sent to it by the ULI. The ULI's buffering of data allows instantaneous data rates to significantly exceed this limit for brief periods.

Read:

The command to read the [T] register is

`T<CR>.`

The ULI will return the following:

aaaaaa<CR><LF>

where: aaaaaa = the 3-byte contents of the [T] register in the range: <000000h : FFFFFFh>.

Write:

To set the value of the [T] register, type

Taaaaaa <CR>

where: aaaaaa = the 3-byte contents of the [T] register in the range: <000000h : FFFFFFh>.

When setting the time base, a 3-digit hexadecimal number must be entered. An error will result if any of the characters is a space or any of the characters is outside the range for hexadecimal characters.

The [T] register specifies the sampling increment the time between measurements in several of the data collection modes. Here are some handy [T]-settings that you may want to use. These values are for the default setting of the [E] register, 00h, which the ULI interprets to be 100h and which corresponds to a clock time unit of 256 µs.

With [E]-setting = 0 (calculated as 256 µs)

T-setting	time (s)
T000027	0.01
T000187	0.1
T000F42	1.0
T009896	10.0

Resetting the [E] register will change the [T]-settings corresponding to various real time. This is sometimes useful. The tables below show [T] values to use when the [E] register has be reset.

With [E]-setting = FAh (250 µs)

T-setting	time (s)
T000028	0.01
T000190	0.1
T000FA0	1.0
T009C40	10.0

With [E]-setting = C8h (200 µs)

T-setting	time (s)
T000032	0.01
T0001FA	0.1
T001388	1.0
T00C350	10.0

Appendix D has a table of commonly used [T] register values.

Example:

In this example, the [T] register will be set to 39 (27h). In most data collection modes, this corresponds to a sample rate of about 100.16 samples per second.

```
ULI Rev. 5.20
H4/3>t000027
H4/3>
```

Register V - Sets the Analog Voltage Triggering Levels

The voltage levels for analog triggering are set using the [V] register. This register is used in conjunction with the [I] register.

The analog triggers look at only the upper 8 bits of the data. Remember this when setting the trigger values. If you are going to collect 8-bit data, the trigger values are the same as the raw values you see. If you are collecting 10-bit data with the original ULI, then you must shift your desired trigger value right by two bits to get your trigger value. Similarly, if you are collecting 12-bit data with the ULI_{II}, then you must shift your desired trigger value right by four bits to get your trigger value.

Read:

The command to read the [V] register is

`V<CR>.`

The ULI will return the following:

aabbccdd<CR><LF>

where:	aa	=	the Port 2 High Threshold (trigger if V2 >= aa),
	bb	=	the Port 2 Low Threshold (trigger if V2 <= bb),
	cc	=	the Port 1 High Threshold (trigger if V1 >= cc),
	dd	=	the Port 1 Low Threshold (trigger if V1 <= dd).

Write:

To set the value of the [V] register, type

Vaabbccdd <CR>

where:	aa	=	the Port 2 High Threshold (trigger if V2 >= aa),
	bb	=	the Port 2 Low Threshold (trigger if V2 <= bb),
	cc	=	the Port 1 High Threshold (trigger if V1 >= cc),
	dd	=	the Port 1 Low Threshold (trigger if V1 <= dd).

When setting any voltage level, all four, 2-digit hexadecimal numbers must be entered. An error will result if any of the characters is a space or any of the characters is outside the range for hexadecimal characters.

Example:

Say you want to set the trigger level of Port 2 of the ULI_{II} to 2.50 volts. You first calculate the digital equivalent value of that voltage. Remembering all ULI's have a maximum input range of 5.12 volts and that the ULI_{II} uses a 12-bit converter, the digital equivalent value for the voltage is

value = (2.50 * 4096)/5.12 = 2000d (07D0h).

The triggering level uses only the upper 8-bits of this equivalent value so this value must be divided by 16 to get the trigger level. In this case, 2000 / 16 = 125d (7Dh).

Incidentally, this is the same level that would be calculated for the original ULI since its Analog to Digital converter has 10-bit resolution. value = (2.50 * 1024)/5.12 = 500d (1F4h). The trigger level is then 500 / 4 = 125d (7Dh).

The trigger level is now applied to the [V] register.

```
...
H4/3>v7d
H4/3>
...
```

Control Commands

Control Commands are used to control how data are sent from the ULI and how the ULI RAM is used. They also allow you to read the value of the ID voltage.

A Control - Device ID Value

The [A] control reads the voltage on the ID input line on the DIN or RJ-11 connectors and returns the results. This register is normally used to test the type of sensor is connected to the port. The concept of ID voltages was originally defined by TERC in their document entitled “MBL Interface Standards.”[7] Several sensors have been designed for use with the ULI that have a network of resistors included inside them to produce a “signature” voltage. This signature voltage can be used as a way of making sure that the right sensor is connected before starting an experiment.
Using this command will disrupt programmable amplifiers so use it before setting up programmable amplifiers.

Read:

The command to read the [A] control is

`A<CR>.`

The ULI will return the following depending on the value of the [S] register:

aaaa<CR><LF>

where:	aaaa	=	the ID code of the device in the active port,
	bbbb	=	the ID code of the device in Port 2 (if [S] register = 2),
	cccc	=	the ID code of the device in the AUX Port (if [S] register = 3).

The result of reading the [A] control will always be reported in hexadecimal format regardless of the display mode.

Example:

The [A] command is issued with a Force Probe connected to Port 1 and a standard Motion Detector connected to Port 2. The ID value for a Force Probe is 027D and for the Motion Detector, 01F7. These two numbers will have slight variations with different Force Probes and Motion Detectors.

```
ULI2 Rev. 1.00
H4/3>a
027D01F7
```

---

[7] In this document, TERC defined a series of resistor ratios that develop an I.D. voltage as a percentage of the supply voltage (Vcc). This was workable when used with TERC's Red Box, since its A/D made **ratiometric** measurements. The ULI, however, uses its A/D to measure **absolute** voltages. Simply put, this means that the I.D. voltage as read by the ULI will vary with the supply voltage. Voltages read will usually be slightly higher (by 1 or 2 counts) than those specified by the MBL standard. The Host program should take this into account by guard banding the target voltage.

B Control - 8-bit Binary Display

The [B] command sets the ULI output format switch for binary display. This type of display is normally used only in programs. **It is very important that XON/XOFF processing is turned OFF for incoming data when using binary mode.** If a byte that looks like an XOFF is received, data will not be accepted by the ULI until an XON is sent to clear it.

After the [B] command is issued, the first character in the ULI prompt will be a “B”.

Write:

To set the display mode to binary, type

B <CR>.

All data returned by a collection mode will be the binary representation of the data, not the characters “0” or “1”. These values may not be printable if a terminal program is being used to collect data. Therefore, this mode should only be used in programs that expect to read binary data.

D Control - Decimal Display

This command sets the ULI output format switch for decimal display and returns the ASCII code of the delimiter and the number of complete records per line. The decimal display format has several options, so the commands to control and check its status are more complicated.

Data output in the decimal mode has leading zeroes suppressed. Note that this means you cannot determine where one decimal field ends and the next one begins just by position of the character in the output string. Using hexadecimal output, character position can be used to determine where one field ends and the next begins.

Read:

The command to read the [D] control is

`D<CR>.`

The ULI will return the following:

aabb<CR><LF>

where: aa = the hexadecimal code for the delimiter <00 : FFh>,
bb = the number of records displayed per line, <01 : FFh>.

The result of reading the [D] control will always be reported in hexadecimal format regardless of the display mode.

Write:

To set the set the display mode to decimal, type

D <CR>.

To set the set the display mode to decimal, delimiter and number of records per line, type

Daabb <CR>.

where: aa = the hexadecimal code for the desired delimiter <00 : FFh>
bb = the number of complete records to be displayed per line <01h : FFh>. You may specify up to 255 complete records of data per line.

The table below lists the ASCII codes of some characters which are commonly used as delimiters.

Character	ASCII Code (decimal)	ASCII Code (hexadecimal)	Character name
<TAB>	09	09	tab
,	44	2C	comma
<SPACE>	32	20	space

<LF>	10	0A	line feed
<CR>	13	0D	return or enter

No filtering is done on your choice of delimiter; i.e., it is the user's responsibility to insure that no illegal characters are selected. For example <Ctrl>+S, <Ctrl>+Q, <Ctrl>+C, etc. are usually used for flow control and thus should not be used as delimiters.

The instruction parser looks for the “D” for decimal mode and then looks for 4 valid hexadecimal characters. If the data following the “D” does not fit the template, it is rejected and the string is treated the same as “D<CR>”.

Example:

For example, Mode 1 reports 2 fields per record – the data field (cccc) and the status field (dd). In decimal mode with a “,” for delimiter and 4 for line length, the ULI would display

cccc,dd,cccc,dd,cccc,dd,cccc,dd<CR><LF>

where cccc = the data field and is from (1) to (10) digits in length and dd is the status field. Note that there are four values of the data field and four values of the status field returned in this example.

In this example, we specify that each decimal output value should be separated by a tab (ASCII code 09h) and only one complete set of measurements from the ULI should be displayed per line on the screen. This effect is demonstrated by reading times in Mode 1.

```
H4/3>d0901
D4/3>m1
Data:
17   1
2882498 3
2905987 1
3109023 3
3123343 1
3285686 3
3308328 1
```

G Control - Global Record

The [G] control provides a quick way to check the status or set the status of a number of ULI commands in one quick step. It is primarily used in programming advanced ULI applications.

The Global record is a collection of 9 registers. The format of the [G] command is

aabbccddeeffgghhiijjkkllmm

where:	aa	256 - the [E] register	
	bb	a reserved byte (do not use)	
	ccddee	the timer [T] register	
	ff	HtrigH	(see [I] register for details)
	gg	HtrigL	(see [I] register for details)
	hh	LtrigH	(see [I] register for details)
	ii	LtrigL	(see [I] register for details)
	jj	the [C] register	
	kkll	Slope an InEnb	
	mm	the [S] register	

Read:

The command to read the [G] control is

`G<CR>.`

The ULI will returns the 13-byte, 26-character string described above. These characters will always be reported in hexadecimal format regardless of the display setting.

Write:

The command to set the [G] control is:

Gaabbccddeeffgghhiijjkkllmm<CR>

When using this command, be careful not to change bytes that are used by the ULI itself. The easiest way to do this is to read the contents of the [G] control, and use a Boolean operation to change only the bytes you need to change.

Example:

In this example, the [G] control is examined to determine its current state. The [E] and [T] registers are then changed and the [G] control is examined again to verify the changes. Finally, the [E] register is changed by changing the [G] control and verified by examining the [E] register.

```
H4/3>g
0000000000000000000F000003
H4/3>efa
H4/3>tfffeee
H4/3>g
0600FFFEEE000000000F000003
H4/3>gfa00fffeee000000000f000003
H4/3>e
06
H4/3>
```

H Control - Switch Display to Hexadecimal

This command sets the ULI output format switch for hexadecimal display.

Write:

To set the display mode to hexadecimal, type

H <CR>.

The only characters displayed are “0” through “9” and “A” through “F”. After the [H] command is issued, the first character in the ULI prompt will be a “H”.

/ Control - Selects Normal Buffer Operation

Normal buffer operation means the RAM pages (as set by the [R] register) are used in a ring buffer configuration for communications. As data are collected, it is placed in this ring buffer until transmitted to the computer. When the ULI starts up, the Normal Buffer Operation is set and the number of RAM pages is set to 1.

Write:

To set normal buffer operation, type

/ <CR>.

After the [/] control is issued, the fourth character in the ULI prompt will be a “/”.

\ Control - Disables Normal Buffer Operation

Normal buffer operation means the RAM pages (as set by the [R] register) are used in a ring buffer configuration for communications. As data are collected, it is placed in this ring buffer until transmitted to the computer. Disabling the Normal Buffer Operation is usually done to have the ULI report data in real-time, i.e., no data buffering. If data are collected before the last data are transmitted, the previous data are overwritten by the latest data. This condition will set the overflow bit in the [J] register.

Write:

To disable normal buffer operation, type

\ <CR>.

After the [\] control is issued, the fourth character in the ULI prompt will be a “\”.

! Control - Selects Flash Mode

If the buffer is set for flash mode, the portion of the buffer reserved will be filled with data and then all of the data are sent to the computer. This setting allows the ULI to collect a large quantity of data very quickly and then send the values to the computer. The rate of data collection is on the original ULI always 13,600 samples per second while the rate on the ULI$_{II}$ is always 11,000 samples per second. The reason the sample rate of the ULI$_{II}$ is slower is that it takes longer to convert a voltage to a 12-bit number than a 10-bit number. If triggering is set, 256 points of pretrigger data are collected.

When the ULI is set for Flash mode, data transmitted by the ULI is always in hexadecimal format regardless of the display settings. The setting of the [E] register controls the sample time for data collection.

Write:

To set the buffer for flash mode operation, type

! <CR>.

After this command is issued, the fourth character in the ULI prompt will be a “!”.

* Control - Selects the Burst Mode

If the buffer is set for Burst mode, the portion of the buffer reserved will be filled with data, data collection is paused, all of the data in the buffer is sent to the computer then, data collection resumes. The rate of data collection is set by the value of the [E] command.

If triggering is set, 256 points of pretrigger data are collected.

When the ULI is set for Flash mode, data transmitted by the ULI is always in hexadecimal format regardless of the display settings. The setting of the [E] register controls the sample time for data collection.

Write:

To set the buffer for burst mode operation, type

* <CR>.

After this command is issued, the fourth character in the ULI prompt will be a “*”.

Example:

This mode might be used to collect statistically noisy data. In this mode, you might collect several hundred quick samples that will be averaged into one reported sample.

<Ctrl>+C

<Ctrl>+C is used to tell the ULI to stop whatever it is doing and get ready to accept more commands. Generally, <Ctrl>+C is used to stop a data collection mode.

This is one of those few commands that are case sensitive. If you are in a terminal program and try to stop a collection mode by holding down the <Ctrl> key and pressing the <C> key, you may find you will also have to hold down the <Shift> key at the same time to stop the collection mode.

<Ctrl>+S

<Ctrl>+S is used to interrupt data transmission. This is also the XOFF character.

<Ctrl>+Q

<Ctrl>+Q is used to resume data transmission after a <Ctrl>+S is issued. This is also the XON character.

ESCAPE Sequences

Escape sequences have a priority lower than the control signals <Ctrl>+Q, <Ctrl>+S and <Ctrl>+C but, have a higher priority than the command line processor. This was done in anticipation of the fact that most host software will be multitasking, and that control sequences can be (will be) issued asynchronous; i.e., without regard to the main (control) activities. The escape processor is prioritized in such a way that flow and break controls will not cause it to lose phase.

These commands are case sensitive; for example, "<ESC> a" will have no effect. This is one of the few times case makes a difference when controlling the ULI.

Escape codes may be issued at the ULI prompt or while collecting data.

The last two escape codes control the tri-state buffer which controls how the multi-purpose pins on the modular telephone connector and DIN connector function.

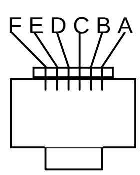
<ESC> A = Digital Out 0 Output High
<ESC> B = Digital Out 0 Output Low
<ESC> C = Digital Out 1 Output High
<ESC> D = Digital Out 1 Output Low
<ESC> E = Digital Out 2 Output High
<ESC> F = Digital Out 2 Output Low
<ESC> G = Digital Out 3 Output High
<ESC> H = Digital Out 3 Output Low
<ESC> J = Sets the ID Voltage/ Digital Out pins to act as Digital Outputs
<ESC> K = Sets the ID Voltage/ Digital Out pins to act as Analog Inputs

An <ESC> followed by any other character (except the control characters mentioned above) will be ignored.

ULI Connectors

RJ-11 Sockets

The ULI has two data port RJ-11 connectors. The original ULI also has one additional connector labeled AUX for auxiliary port. The three RJ-11 sockets are located at the lower-right corner of the original ULI. Two RJ-11 sockets are located on the front panel of the ULI$_{II}$. The modular phone connector ports have the pin assignments shown below. This is a diagram looking into the socket on the ULI.



Pin Assignments for the Phone (RJ-11) Jacks

Pin	Port 1	Port 2	AUX[8]
ID input/ Digital output	Analog In 10/ Digital Out 0	Analog In 7/ Digital Out 1	Analog In 6
Ground	Ground	Ground	Ground
+5 Volts	+5 Volts	+5 Volts	+5 Volts
Analog input	Analog In 9	Analog In 8	Analog In 4
Digital output	Digital Out 2	Digital Out 3	Digital Out 2
Digital input	Digital In 0	Digital In 1	

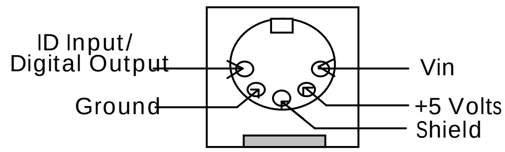
This connector follows a standard established by Technical Education Research Centers. The ID and analog inputs have an input range of 0 to +5.12 volts. The multi-function pin (pin A) can be configured for use as an additional

[8]Available on the original ULI only.

analog voltage input line or a digital output line. The +5 volt lines are loosely regulated which means you should not use these as reference voltages. Several existing probes and sensors can be used with it. The ID input is for an identifying voltage which allows the ULI to recognize the type of sensor connected. Probes using this connector include the Force Probe, and the Current and Voltage Probes.

DIN Sockets

Two 5-pin DIN plugs are located at the lower-right side of the original ULI. Four 5-pin DIN plugs are located on the front panel of the ULI$_{II}$. These connectors are designed for use with sensors which produce a voltage signal. The pin configuration for the 5-pin DIN socket is shown here:



Pin Assignments for the 5-Pin DIN Sockets

Pin #	DIN 1	DIN 2	DIN 3[9]	DIN 4[10]
ID Input/ Digital Output	Analog In 10/ Digital Out 0	Analog In 7/ Digital Out 1	Analog In 4/ Digital Out 4[11]	Analog In 1/ Digital Out 5[12]
Ground	Ground	Ground	Ground	Ground
Shield	Ground	Ground	Analog In 5[13]	Analog In 2[14]
+5 Volts	+5 Volts	+5 Volts	+5 Volts	+5 Volts
Vin	Analog In 9	Analog In 8	Analog In 6	Analog In 3

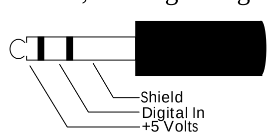
The voltage range of the Analog Inputs is 0 to +5.12 volts. The multi-function pin can be configured for use as an additional analog voltage input line or a digital output line.

The DIN sockets are used for many of the sensors sold for use with the ULI such as the test leads provided with the ULI, pH probes, and thermocouples. These connectors are also recommended for circuits that you build yourself to connect to the ULI. Several “do-it-yourself” circuits are described in the manual which comes with the Data Logger program. The 5-pin (180°) DIN plug for connection to this connector is available at most electronics stores, including Radio Shack.

This DIN socket uses the same pin out pattern as the Vernier Software Voltage Input Unit, Serial Box Interface, and MultiPurpose Lab Interface. All Vernier Software circuits and projects which use 5-pin DIN plugs can be used with the ULI via this connector.

1/4-inch Phone Sockets (ULI$_{II}$ only)

Two 1/4” Phone sockets are located on the front panel of the ULI$_{II}$. These inputs are normally used for photogates and Radiation Counters. As shown below, the tip is +5volts, the ring is Digital in and shield is ground.



Connector Assignments

Connector	DG1	DG2
Shield	Ground	Ground

[9]Available on the ULI$_{II}$ only.

[10]Available on the ULI$_{II}$ only.

[11]This digital out line cannot be controlled by the user.

[12]This digital out line cannot be controlled by the user.

[13]Connected to ground through a 1-megaohm resistor.

[14]Connected to ground through a 1-megaohm resistor.

Digital In	Digital In 0	Digital In 1
+5 Volts	+5 Volts	+5 Volts

The signal can be either ground or +5 volts without damaging the ULI. These inputs use 74HC Schmitt Trigger logic. The digital inputs are tied to ground via a 6.8K ohm resistor.

RCA Connectors (Original ULI Only)

The original ULI has a row of RCA phone jacks along the front for use with a wide variety of sensors and probes. Each of these connectors is described in this section. These connectors can be used with RCA phone plugs, which are available at any electronics store.

RCA Power

The two jacks on the left side of the ULI are for supplying power to photogates or other sensors. The center terminal is +5 volts DC. The outer connector is ground. The current draw from the +5V line should be limited to 20 milliamperes sustained, 50 ma maximum.

RCA Digital In

There are two digital input lines on the ULI. The center of these jacks is the input and the outside is ground. The input line can be shorted to either ground or +5V without damaging the ULI. These inputs use 74HC Schmitt trigger logic. These connectors are commonly used for photogates and the Radiation Monitor. The digital input lines are tied to +5 volts via a 10K ohm resistor.

RCA Voltage In

The center terminals of the Voltage In phone jacks can be read by the Analog-to-Digital converter. The voltage range is 0 to 5.12 volts DC. The voltage is converted to a count which has a 10-bit resolution. The outside connector on these jacks is grounded. These inputs are electrically equivalent to the analog input on the RJ-11 connector and the Vin lines of the DIN connectors.

RCA Digital Out

The center terminal of each of these two RCA jacks can be set to either high (5V) or low (0V) by commands issued to the ULI. These are totem pole CMOS 74HC outputs. The outside of these jacks is +5V.

RCA Resistive In

These two terminals can be used with sensors that vary in resistance. They are similar to the “stick” inputs on an IBM game port or the PDL inputs on an Apple II game port. They can be used to measure resistance in the range 1 K• to 1 M•. The outside of this connector is +5V. The internal capacitor that is associated with these inputs is 0.01 µF.

Analog Port (Original ULI only)

This connector is not frequently used, but it can provide access to all of the analog voltage input lines on the ULI. It can be used for diagnostic tests on the ULI and in situations where users want to monitor many different analog input signals. The chart below shows the connections on the 16-pin DIP analog port socket:

1	power (5 V)	16	ref. voltage (5.120V)
2	analog voltage ch. 0	15	A/D chip select
3	analog voltage ch. 2	14	analog voltage ch. 1
4	analog voltage ch. 4	13	analog voltage ch. 3
5	analog voltage ch. 6	12	analog voltage ch. 5
6	analog voltage ch. 10	11	analog voltage ch. 7
7	analog voltage ch. 9	10	analog voltage ch. 8
8	analog ground	9	digital ground

Cal + and Cal - (Original ULI only)

Model 4C (and newer) of the original ULI’s have 2 pins near the Analog Port labeled "Cal +" and "Cal -". There is a 40 mV (+/-0 1%) voltage across these pins which can be used to calibrate external amplifiers, pre-scalers, etc.

Expansion Port (Original ULI only)

The Expansion Port is a 16-pin DIP socket that provides access to a wide variety of lines. The chart below shows the connections on the 16-pin DIP analog port socket:

1	resistive input ch. 4	16	power (5 V)
2	resistive input ch. 2	15	resistive input ch. 3
3	digital output ch. 2	14	resistive input ch. 1
4	analog voltage ch. 7	13	digital output ch. 1
5	analog voltage ch. 8	12	analog voltage ch. 10
6	digital input ch. 2	11	analog voltage ch. 9
7	no connection	10	digital input ch. 1
8	system ground	9	analog ground

Multiple Connectors

Note that several input lines on the ULI extend to more than one connector. Trying to read a signal from one of these inputs when more than one input device is connected to it can cause very confused results. One example would be using a ULI Force Probe connected to the Port #1 phone connector while a voltage probe was connected to DIN #1 connector.

DB-25 Socket

The ULI uses a 25-pin D socket for the serial connection. The connector uses a standard RS-232 pin configuration. In communications protocol, the ULI is considered a “Data Set.” The serial cable for use with the ULI is the same as one that would be used with a modem. The pin out of the ULI 25-pin serial port socket is described below:

RS-232C DB25 Pin Assignments

Pin	Name	Function
1	FG	Shield (Frame Ground)
2	RxD	Receive Data
3	TxD	Transmit Data
4	CTS	Clear To Send
5	RTS	Request To Send
6	DSR	Data Set Ready
7	GND	Signal Ground
8	DCD	Data Carrier Detect
20	DTR	Data Terminal Ready

Pins 4 and 5 are connected inside the ULI. Pins 6, 8, and 20 are also connected internally. They are not used by the ULI. The minimum cable requirements are pins 2, 3 and 7.

SPI Socket (ULI_{II} only)

This socket is currently used for testing purposes only.

Power Connector

The connector labeled 9V/1A is a 2.1-mm DIN power jack for the power adapter. A 9-volt (at full load) adapter capable of supplying up to 1 ampere should be used. The center connects to the negative lead. Power adapters for connection to this port are readily available. The ULI contains circuitry to protect it from reverse polarity, but it is not fused internally. The power adapter used should be self-limiting. All UL approved, wall-mount DC adapters should meet this requirement.

Since the ULI uses a power adapter rather than a built-in power supply, it is easy to use in parts of the world outside North America. In countries with electrical power lines with different voltages and frequencies, all that is needed is a power adapter that works with the local power lines and provides 1 amp at 9VDC.

The Communications Bottleneck

While the ULI can theoretically communicate with its host computer at almost 4000 characters per second (at 38,400 baud), the actual rate at which it sends data to the host is usually much lower. In fact, the host computer "paces" the data transfer from the ULI to match its ability to receive and display the information the ULI is sending. With a slow host computer, the actual data transfer rate can be as low as 40 characters per second (even if the nominal data transfer rate is set to a much higher value such as 9600 baud). Since many modes send 10 or more characters for each data logging event, this limits the rate at which you can log data to your computer screen to no more than a few events per second.

Fortunately, this is not the end of the story. The ULI has an internal buffer which can record data at rates of up to about 400 measurements per second using Ring buffer setting. You can collect data much faster using Flash and Burst modes. In these modes, when the buffer is full, data logging can be automatically terminated and the ULI will then send the data in its buffer to the computer. The ULI offers three distinct buffering modes, termed *ring*, *flash*, and *burst*. (A fourth data transmission mode is also available. This is the unbuffered, real-time mode set by the [/] register.) In each of the three buffered modes, the size of the buffer is set using the [R] register. Use the command Rnn command, where nn is the number of 256-byte "pages" of buffer space set aside for data logging. The range of nn is <1h: 20h> and its default value is 1. Before increasing the buffer size to some large value, you should think about how much data you really want. Reading large amounts of data can be slow to transfer to the computer.

When you are using the ring buffer to log data at rates higher than your host computer can digest them, you will want to stop the data logging when the buffer is full. This happens automatically in the Flash and the Burst modes, but not in the Ring mode. To prevent the ring buffer from overfilling, just issue the command J40 before initiating data logging. When the buffer is full, data logging will be terminated automatically and the message "Err0" will be appended to the end of your data set if all is well.

Typing a <Ctrl>+C at any time will terminate data logging to the buffer. Remember that data will continue to be transferred from the buffer to the host until the buffer is empty, even after you (or the ULI) terminate data logging.

Here is a simple example of how to log at least 256 values of a voltage signal connected to port 1 at a rate of 1000 samples per second with 10-bit resolution on the original ULI (or 12-bit resolution on the ULI$_{\text{II}}$):

M0<CR><SPACE>	[resets the ULI]
D<CR>	[selects digital output]
S1<CR>	[selects port 1 only]
C2<CR>	[selects 2 bytes per measurement (for 10-bit resolution)]
EFA<CR>	[sets the clock to 250 µs per tick]
T000004<CR>	[makes a measurement every 4 ticks (= 1 ms) or 1000 per second]
J40<CR>	[stops when the buffer is full]
R02<CR>	[reserves a 512-byte buffer to hold 256 2-byte measurements]
M8<CR>	[initiates data logging]

Appendix A - Technical Information

Microprocessor: SAB A-P, 8032 running at 12 MHz – includes 256 byte of internal RAM, four 8-bit ports and three 16-bit timers.

EPROM: 27C64, 8k bytes programmed with routines to control data collection.

Analog-to-Digital Converter: The ULI$_{II}$ uses the TLC2543IN - 12-bit, switched-capacitor, successive-approximation A/D converter. The original ULI uses the TLC1541IN - 10-bit, switched-capacitor, successive-approximation A/D converter. Each of these ADC's have 11 inputs and an on-chip multiplexer. An internal self-test voltage is provided. Sample-and-hold circuitry is included in the IC.

Voltage input range:	0 to 5.12 volts
Input Impedance:	1012
Conversion time:	21 µs
Channel acquisition time:	5.5 µs
Maximum samples/sec:	32k
Total unadjusted error:	+/- 1.0 LSB maximum

RAM: SRM2264, 8k bytes, can be used for temporary data storage.

Digital Inputs: Two 74HC Schmitt trigger logic inputs.

Digital Output: Two totem pole CMOS 74HC outputs.

Serial Communication: Maximum baud rate: 38.4k (The ULI automatically matches the baud rate being used by the computer)

Word Length: 8 bits
Parity: none
Stop Bits: 1
Handshake: XON/XOFF (<Ctrl>+S, <Ctrl>+Q)

Power Supply: 9-volt, 1 amp, wall converter

Connectors on Original ULI:

- (3) 6-pin modular telephone connectors including voltage in, digital output, digital input and power leads.
- (2) 5-pin DIN sockets including voltage in and power leads.
- (2) Power RCA jacks
- (2) Digital In RCA jacks
- (2) Voltage In RCA jacks
- (2) Digital Out RCA jacks
- (2) Resistive In RCA jacks
- Power jack
- RS-232 socket

Connectors on ULI$_{II}$:

- (2) 6-pin modular telephone connectors including voltage in, digital output, digital input and power leads.
- (4) 5-pin DIN sockets including voltage in and power leads.
- SPI port
- Power jack
- RS-232 socket

Appendix B -Troubleshooting Guide

Problem	Cause	Solution
The computer does not display any responses from the ULI	Serial communication cable is not set up properly or is faulty	Make sure that the cable you are using is a “modem cable.” Pins 2, 3 and 7 of the ULI serial connector should be connected to the corresponding pins on the serial port of the computer.
	Wrong serial port being used	Make sure you are using the correct serial port. For example, on an IBM compatible don’t confuse COM1 and COM2. On a Macintosh don’t confuse the modem port and the printer port.
	Wrong communication protocol is being used	Make sure that the computer is set for word length=8 (8 data bits), no parity, 1 stop bit.
	ULI is not turned on	Turn on ULI. The Red LED should be on.
	ULI is not receiving power	Make sure that the power adapter is plugged in and connected to the ULI.
	ULI is not correctly connected to Macintosh	Connect ULI using Macintosh modem cable to the modem port.
Responses of the ULI are unreadable and include strange characters	ULI is set for binary	Type “H” or “D” to change to hexadecimal or decimal.
ULI not communicating with host computer	Baud rate or other communications protocol breakdown	Turn the ULI off and then back on and start again.
The ULI continues to send data even after the host computer sends <Ctrl>+C	The ULI’s data collection rate exceeded the communications rate through the serial port	Take data at a slower rate, or be patient while the ULI sends its buffer contents to the computer
	A lower-case C is being used	Some computers require that the C be in upper case
Readings are noisy	ULI is picking up interference signals from the computer	Place the ULI at least 30 cm from the computer.

Appendix C - Summary of Commands

A Summary of ULI Commands

Control Commands: (All values returned are in hexadecimal notation.)

A	reports the ULI reference voltage used to identify probes
B	sets the display switch to return data in 8-bit binary
Daabb	sets the display switch to return data in ASCII decimal with each field followed by a delimiting character (ASCII code in hexadecimal = aa), bb full records per line
G	reports the status of the [G] register (13 bytes)
H	sets the display switch to return data in ASCII hexadecimal followed by <CR><LF>
/	selects normal (ring) buffer operation
\	disables the buffer
!	selects flash mode, collecting one RAM full of data and transmitting when done
*	selects the burst mode, collecting RAM full of data and transmitting, then repeating

Register Commands: (All arguments all are in hexadecimal format.)

Cn	sets data word length to n bytes (1, 2, 3, or 4 bytes)
Eaa	sets the time base <01 : FFh>; 00h means 100h which equals 256 µs
Iaa	sets the digital and analog triggering conditions
Jaa	sets the mask for error trapping
Kaa	sets the amplifier settling time register
Paaa	sets the gain of a programmable amplifier
Paaabbb	sets the offset and gain of a programmable amplifier
Raa	specifies the number of pages of RAM to be reserved for the buffer <01 : 20h>
Sa	selects active port(s) n: 1, 2, 3 (= both), 4 (for AUX)
Taaaaaa	sets timer to aaaaaa(hexadecimal) * time base ([E] register) in µs <000001:FFFFFFh>; 000000h means 1000000h
Vaabbccdd	sets the analog voltage triggering levels

Miscellaneous Commands:

<Ctrl>+C	stops current operation and return to command mode
<Ctrl>+S	interrupts data transmission
<Ctrl>+Q	resumes data transmission
<ESC>A	sets digital output 1 high
<ESC>B	sets digital output 1 low
<ESC>C	sets digital output 2 high
<ESC>D	sets digital output 2 low
<ESC>E	sets auxiliary output 1 high
<ESC>F	sets auxiliary output 1 low
<ESC>G	sets auxiliary output 2 high
<ESC>H	sets auxiliary output 2 low
<ESC>J	sets tri-state enable
<ESC>K	sets tri-state disable
M0	resets ULI (must be followed by a press of the <SPACE BAR>)
MFF	diagnostic test of the ULI

Data Collection Commands:

Mode 1	reports the time at which any digital input port changes its state
Mode 2	determines echo times from the motion detector
Mode 3	reports electrical resistance (Used on the original ULI only)
Mode 4	reports period of the signal on digital input 1
Mode 5	reports each complete pulse width (in µs) at one digital input
Mode 6	stopwatch emulation
Mode 7	reports the pulse count at one digital input during each preset timer interval
Mode 8	reports, after each preset timer interval, values of one or both voltage inputs
Mode 9	runs the sonic ranger and the Force Probe together

Mode A	reads up to 14 analog voltages once each preset time interval
Mode B	reports electrical resistance and analog inputs
Mode C	reports the status of the digital outputs and the analog voltages inputs
Mode D	sequential pulse period timer
Mode E	reads rotary motion detector
Mode F	reads rotary motion detector and analog inputs

Appendix D - [T] Register Settings

The [T] register specifies the sampling increment the time between measurements in several of the data collection modes. Here are some handy [T] register settings that you may want to use.

With the default [E]-setting = 0 (calculated as 256 µs)

[T] register	Time (s)	Freq (Hz)
T000027	0.00998	100.16
T00004E	0.02022	49.446
T0000C3	0.04992	20.032
T000187	0.10010	9.9904
T00030D	0.20019	4.9952
T0007A1	0.49997	2.0001
T000F42	0.99994	1.0001
T001E84	1.99987	0.5000
T004C4B	4.99994	0.2000
T009896	9.99987	0.1000
T039387	60.0000	0.0167

Resetting the [E] register will change the [T]-settings corresponding to various real time. This is sometimes useful. The tables below show [T] values to use when the [E] register has been set for a different timebase.

With [E]-setting = FAh (250 µs)

[T] register	Time (s)	Freq (Hz)
T00000A	0.0025	400
T000019	0.005	200
T000028	0.01	100
T000050	0.02	50
T0000C8	0.05	20
T000190	0.1	10
T000320	0.2	5
T0007D0	0.5	2
T000FA0	1.0	1
T004E20	5.0	0.2
T009C40	10.0	0.1

With [E]-setting = C8h (200 µs)

T-setting	time (s)	freq (Hz)
T000019	0.005	200
T000032	0.01	100
T000064	0.02	50
T0000FA	0.05	20
T0001FA	0.1	10
T0003E8	0.2	5
T0009C4	0.5	2
T001388	1.0	1
T0061A8	5.0	0.2
T00C350	10.0	0.1

Appendix E - Sample Programs

These simple programs are meant to help get you started at programming for the ULI.

QBASIC or QuickBasic on an IBM-compatible computer.

This program will send a <SPACE BAR> character to the ULI, set the [T] register for 1 reading per second, read a few voltages from the Voltage In ports, and report the counts on the computer screen.

```
100 ‘  ULI_D.BAS - READ VOLTAGES FROM ULI
110 ‘
120 XOFF$=CHR$(19):XON$=CHR$(17):  ‘ XON/XOFF CHARACTERS
130 COMFIL$=“COM1:9600,N,8,1,DS0”: ‘ SET UP COMM PORT
140 OPEN COMFIL$ AS #1:     ‘ OPEN COM1 TO ULI
150 PRINT #1, “ “;:      ‘ SEND SPACE TO ULI
160 GOSUB 800:      ‘ DELAY
170 GOSUB 500:      ‘ READ PROMPT
180 GOSUB 800:      ‘ DELAY
190 PRINT #1, “T000F42”:    ‘ ONE SECOND PER READING
200 GOSUB 800:      ‘ DELAY
210 PRINT #1, “D”:      ‘ DISPLAY IN DECIMAL MODE
220 GOSUB 810:      ‘ DELAY
230 PRINT #1, “M8”:      ‘ READ VOLTAGES
240 GOSUB 500:      ‘ READ FIRST VALUE
250 FOR N = 1 TO 5:      ‘ READ FIVE TIMES
260 PRINT “ULI READING #”;N;”: “;  ‘ DISPLAY READING NUMBER
270 GOSUB 500:      ‘ READ VALUE FROM ULI
280 NEXT N
290 END:        ‘ END PROGRAM
500 ‘
510 ‘READ FROM ULI AND DISPLAY IT ON THE SCREEN
520 ‘
530 IF EOF(1) THEN 530:    ‘ WAIT HERE FOR RESPONSE
540 GOSUB 810:      ‘ DELAY
550 WHILE NOT EOF(1)
560 A$=INPUT$(LOC(1),#1):   ‘ READ IN A CHARACTER
570 WEND
580 PRINT A$;:      ‘ PRINT THE CHARACTER
590 RETURN
800 ‘
810 ‘ DELAY SUBROUTINE
820 ‘
830 D=1000       ‘ SET LENGTH OF DELAY
840 FOR J=1 TO D: NEXT J    ‘ LOOP FOR DELAY
850 RETURN
```

Turbo Pascal on an IBM-compatible computer

This program activates the ULI and places the ULI in the self-test state using the MFF mode.

```
PROGRAM QuickTest;
{ Quickly tests the ULI by issuing the MFF mode. Demonstrates write only
communication with the ULI. }
USES
Crt;         { Turbo Pascal screen handling }
VAR
ULI : TEXT;       { Address ULI as a text file }
BEGIN { QuickTest }
Write (‘Turn the ULI off then on... Press <ENTER> to continue.’);
ReadLn;        { Wait for user to press enter }
Write (‘Testing ULI... Press <ENTER> to exit.’);
 Assign (ULI, ‘COM1’);    { Assign COM1 to ULI }
 ReWrite (ULI);      { Open COM1 for write only }
 Write (ULI, ‘ ’);     { Send space to ULI }
 Delay (500);       { Let it display its prompt }
 WriteLn (ULI, ‘MFF’);    { Start the ULI test mode }
 ReadLn;        { Wait for user to press enter }
 WriteLn (ULI, ^C);     { Stop the ULI test }
 Close (ULI);       { Close COM1 }
END. { QuickTest }
```

Appendix F - Tips for Programmers

- Controlling the serial port can be fairly tricky in some programming languages. Refer to your manuals and sample code segments. A good way to study exactly what the ULI is sending back in every situation is to use it with a telecommunications program. In this way you can "simulate" what your program will be sending to the ULI and note exactly how it responds back. We have found this very helpful.

- The correct test to see if a data collection mode started without error is to check the first line against Data:<CR><LF>". If the first line does not match, then there is some problem. If it does, the ULI is collecting data. Note that "Data:" is displayed BEFORE the ULI begins waiting for a trigger event if any triggers are on, which means that a timeout of one second is safe when waiting for the "Data:".

- When using decimal output for any of the microsecond timing modes (1, 6, and D), make sure to set [C] = 4. This will help provide future compatibility.

- Remember to turn off XON/XOFF on the receiving end if you plan to use binary output. Otherwise, every once in a while the ULI will send some raw data that looks to your computer like an XOFF and your computer will stop sending data to the ULI.

Appendix G - Using the ULI with Batteries

As laptop and portable computers have become more popular, several people have contacted us about using the ULI under battery power.

We have powered the ULI from eight NiCad C-cells in series. The resulting voltage is about 9.6 volts. You also can use batteries designed for use with video cameras. A Radio Shack #274-1573 Coaxial DC Power Plug can be used for the connection at the ULI. Note that the plug should be wired center negative.

The currents drawn by the ULI under a variety of conditions are listed below:

On, with no sensors connected	205 ma
With Motion Detector plugged in	240 ma
With Motion Detector running	245 ma
With ULI Force Probe only	210 ma
With Vernier Software pH probe	220 ma
With Vernier Software thermocouple	205 ma

We found that the ULI would run for about three hours on the eight NiCad C cells. It would run a lot longer on alkaline batteries.