Pembina Lake Erie Datalogger Requirements

Preliminary Specification

February 21, 1995

1. Project Overview

1.1 Introduction

Pembina Resources, from hereon referred to as "the company", is involved in the exploration and subsequent commercial exploitation of natural gas reserves in lake Erie. Several hundred wells are currently in operation, producing through a network of underwater pipelines. A group of several wells is called a field. Fields are connected to the main line through junctions. Junctions and wells are equipped with special short sections of pipe called "meter runs", through which all gas flows. These contain an inline, precisely machined orifice, called a "plate" that tends to restrict gas flow and thus introduces a pressure difference across itself. The meters are equipped with fittings, so that pressure readings can be taken immediately before and after the orifice. These are named upstream and downstream pressures respectively. Flow rate can then be calculated, by including these quantities in a standard formula, and is usually expressed in MCF/D or thousand cubic feet per day. Historical data of pressures and flow rates provides valuable information to the field engineer, and thus serves to dramatically enhance the decision process.

1.2 Background

Traditionally, the taking of pressure readings at a site has been done by divers, using a test hose with attached dial pressure gauge. The test hose is lowered to the diver from the diveboat, and pressures are recorded. This is termed an ILOT (Inline Orifice Test). The economics involved prohibit frequent site visits of more than once per month, due to the cost of diveboats and divers. The company has successfully used electronic dataloggers, developed over the last five years, to record pressures and temperature at a frequency of 48 readings per 24 hour period (twice per hour). Due to capability limitations in the devices used (resolution, memory capacity, power consumption) a new datalogger is needed to perform the functions mentioned above and additional functions, yet to be discussed.

1.3 Scope

The following specification attempts to outline general requirements for a desired system as prescribed by experience gained in the field. **This document is not a complete technical specification for the system.** The datalogger and compatible communication software must be supplied. The plotting software is optional. **For the remainder of this document, resources marked with a check mark (☑) are necessary, while the ones marked (☐) are optional.**

- ☑ **Datalogger;** To supply a multi-channel datalogger to be used in the underwater natural gas production industry, capable of recording values from pressure transducers and temperature sensors with the optional capacity to control external devices and communicate with the host computer through a wire or wireless RS232 link. Must be capable of recording up to eight channels, twice per hour, for up to a year. Must have "sleep" function to preserve battery power.
- ☑ **RS232 link;** Communication should be interactive. Ideally, the host issues commands and the datalogger responds. Custom communication software will be accepted as long as communication is possible using a market-standard package.
- ☐ **Plotting software;** Must support HPGL and be capable of plotting up to 20,000 data points on one graph. This allows a years' data to be plotted. Plotted output to optionally contain comments entered in the field, in the manner shown in section 4.3.

2. Datalogger Specification

2.1 General Operation

The datalogger resides in an air-tight plastic container, which is placed in a stainless steel vessel. The top of the vessel contains the pressure transducers (up to four). These are wired to the datalogger through air-tight bulkhead connectors, and are connected to the pressure measuring fittings with flexible hoses. The datalogger will "wake-up" after a predetermined time interval has elapsed, record inputs as per setup and put itself back to "sleep". This ensures that power consumption will be kept at a minimum. Attachment of the communication cable will also result in the logger "waking up". Maximum "on" time to record all channels must not exceed 6 seconds. Here is a list of desired resources:

2.2 Hardware

- ☑ 8 channel multiplexer.
- ☑ Integral 12 bit (plus sign and polarity) or better A/D converter, preferably of the dual slope integrating type. Conversion speed - ten per second.
- ☑ On board real time clock (RTC), so readings in memory can be time stamped and time-based commands can be executed in real time. The RTC is to be powered by the backup supply or have its own supply.
- ☑ Minimum of 6 analogue inputs, (maximum 8), with sufficient range to represent the 0-5 Volt and 4-20 milliamp scales.
- ☑ One RS-232 compatible serial port configured as a DTE (Data Terminal Equipment), such that a standard MODEM can be plugged into it. This port should be capable of all standard baud rates, up to 9600.
- ☑ Battery switch for external devices such as current or voltage output pressure transducers. This allows main power to supply the sensors and transducers for the "on" time of the datalogger.
- ☑ Backup supply for memory retention and other standby and "sleep" functions. This is separate from the main power source (24Volt, 4Amphour Batteries), and supplies the memory backup power.

- ☐ At least one output capable of driving external devices (usually a solenoid valve), drawing up to 1 Amp @ 24 Volts.
- ☐ Differential inputs, with a common-mode range of 6 Volts. Isolation is not required.

2.3 Memory

Datalogger must have enough memory capacity to store up to eight channels, every thirty minutes for up to a year. Additional memory space must be allowed for to store optional comments and time. In the following example each reading requires 16 bits of storage, and readings are to be taken every thirty minutes. This would require 2 bytes X 8 channels X 2/hour X 24 hours X 365 days = 280320 Bytes. Operation should be such, that when memory becomes full, older entries will be overwritten.

2.4 Size

The datalogger and its enclosure must fit in a rectangular box having the following internal dimensions (height, width, length - in inches): 7.5" x 4.0" x 3.0". Approximately one half of this volume is taken up by the wiring harness and by space allocated for future expansion. Consequently, the maximum allowable datalogger dimensions including battery and connectors are: 7.5" x 4.0" x 1.5".

2.5 Firmware

All necessary program data and program code to reside on EPROM or FLASH ROM. All firmware functions listed, to be available to the user through the communication software. These are:

- ☑ Command to **calibrate** an input. The program requires minimum, maximum, range and label of input channel.
- ☑ Command to **read battery** voltage. Returns battery voltage on success, error message on failure.
- ☑ Command to **clear unit memory** and restore setup defaults. Returns amount of available memory on success, error message on failure. Requires master level password.
- ☑ Command to **set unit ID** and serial number. Returns nothing on success, error message on failure. Requires master level password.
- ☑ Command to **read unit ID** and serial number. Returns ID and serial number on success, error message on failure.
- ☑ Command to **set time and date** on the remote RTC. Returns nothing on success, error message on failure. Requires master level password.
- ☑ Command to **read amount of memory** used in the logger.
- ☑ Command to **dump all records** in memory. Returns all memory contents on success, error message on failure.
- ☑ Command to **display all channels in real time**. Returns values of all input channels, then repeats, until interrupted. Returns error message on failure.
- ☑ Command to setup **auto read**. Requires starting time and date, and interval in tenths of a minute. Minimum 0.1 minute, maximum 60 minutes. Returns nothing on success, error message on failure.
- ☑ Command to **setup datalogger**, as described in setup file. Returns nothing on success, error message on failure.
- ☐ Command to **set the output** on or off. Returns nothing on success, error message on failure. Requires operator level password.
- ☐ Command to **delete a range** of records. Returns nothing on success, error message on failure. Requires operator level password.
- ☐ Command to **dump memory contents**, based on date range. Returns specified record range on success, error message on failure.
- ☐ Command to **store a comment** (time stamped). Returns nothing on success, error message on failure. Requires operator level password.
- ☐ Command to **update ROM**, if FLASH ROM equipped. Returns nothing on success, error message on failure.

2.6 Power

The following assumes that the datalogger and external devices are turned on for 10 seconds in a thirty minute period. This represents an "on" time with 0.56% duty cycle. Two power sources supply the system as follows:

Main Power; Usually, 24 Volt, 4 Amphour batteries. These batteries supply the pressure transducers and temperature sensors for a minimum of 12 months. It is acceptable to use this source to supply power to the datalogger, as long as the datalogger has its own, internal power supply.
Datalogger Power; This power source keeps the datalogger alive for a minimum period of twelve months. If voltage falls below a safe value, the device stops operating and uses remaining power to retain memory only. Memory should remain intact for a minimum of five years.

3. Communication Program Specification

3.1 General

The communication program must provide the interface through which commands are sent to the datalogger, as well as keep track of site-pertinent data such as, site ID, number of active channels, factors and calibration values for individual channels, etc. Upon invocation, the program asks for a valid password and presents a list of several sites to select from. Once a site is selected, it connects to the datalogger. When connected, only the commands respective to password level (operator, master), will be available.

3.2 Operation

Difficulty of dealing with stored data and filename contention, increases according to the number of installed dataloggers. The following suggests a logical way to eliminate these problems, **if custom communication software is supplied.** The program depends on the following disk files for operation:

1. **A file with a [.ini] extension, for program setup variables. These are the program directory, datafile directory, and general defaults.**
2. **One or more setup file(s), with [.ste] extension, containing site and channel information.**

Additionally the program should be able to:

- ☑ Create a new setup file
  - Requires master level password
- ☑ Edit an existing setup file
  - Requires master level password
- ☑ Communicate with the datalogger
- ☑ Send a setup file to the datalogger
  - Requires master level password
- ☑ Save datalogger data to disk in an intelligent manner.

4. Plotting Program Specification

4.1 General

Several company requirements necessitate the use of a custom or semi custom plotting software package. Semi custom implies a translator program. Such software should include:

1. **HPGL support for 7475A series of plotters with capability to save output to a file.**
2. **Output to screen (VGA resolution or higher).**
3. **EXCEL® and ASCII output, COMMA or TAB delimited.**
4. **Data plotted on standard "A" or "B" size paper, with a standard title block for pertinent information.**
5. **Up to six data series (u/s, d/s, annulus, casing, temperature, battery).**
6. **Up to three vertical axes (pressure, temperature, voltage).**
7. **Ability to include or exclude title block.**
8. **Ability to rename vertical and horizontal labels.**

4.2 Input

Input to the program in ASCII, as created by the communications program. Here is a conceptual look of what this might resemble:

```
T, 02/12/95, 12:32:00      << Date and time entry
C, 0004                    << Code to indicate four values to follow
1, 134.2                   << First value
2, 130.4
3, 456.6
4, 33.4                    << Last value
T, 02/12/95, 13:02:00      << Date and time entry
A, Plate Change to 0.500   << Alphanumeric comment
```

4.3 Output

Here is an example of desired plotting program output: