Rain Gauge v4.1.1

Operating procedures and theory of operation

Matthew Casari

2016

Table of Contents

[1. Introduction 2](#_Toc446683906)

[2. Theory of Operation 2](#_Toc446683907)

[3. Operating Procedures 3](#_Toc446683908)

[3.1. Electrical Characteristics 3](#_Toc446683909)

[3.2. Physical and Serial Connections 3](#_Toc446683910)

[3.3. User Interface 5](#_Toc446683911)

[3.3.1. Data Retrieval – Direct Measurement Mode 5](#_Toc446683912)

[3.3.2. Data Retrieval – 60 Minute Report 6](#_Toc446683913)

[3.3.3. Input Console 8](#_Toc446683914)

[Appendix A: Console Commands 9](#_Toc446683915)

[Appendix B: List of Acronyms 10](#_Toc446683916)

[Appendix C: Mathematical References 11](#_Toc446683917)

[Statistical Data Pseudo-Code: 11](#_Toc446683918)

[Volume Calculations formula: 12](#_Toc446683919)

# Introduction

In the 1990’s, PMEL developed a voltage-to-frequency interface monitoring the R.M. Young rain gauge capacitive sensor, providing higher resolution rain data. The existing revision of this interface hardware, Rev 4.1, has a machine controlled interface requiring the operator to use an intermediate controller to retrieve data from the sensor. The native data is a pulse-width modulated signal which must be decoded to be used. The existing microcontroller on-board is vintage of the 1990s, and has limited capabilities.

As an update to the existing analog hardware and PWM interface, PMEL has integrated a new microcontroller to replace the existing PIC16 microcontroller. The new microcontroller, the MSP430FR5949, has multiple digital I/O and onboard UART drivers. It is low power, can run on a 32kHz crystal, has an internal oscillator capable of 16MHz, and an onboard Real-Time Clock with user programmable alarms. The MSP430FR5949 also has 64KB of non-volatile FRAM, allowing for a large amount of data to be stored on-board even during power outages.

The user now has the ability to read data over an RS-232 connection, as well as set metadata values stored in memory which allow for conversion of raw data to volumes (mL). Additional function includes the ability to set a real-time clock to verify that the sensor has not lost power (or drifted over time), as well as the ability to report one minute average data for the past 60 minutes.

# Theory of Operation

The new microcontroller interfaces directly with the previous circuitry version frequency signal. The analog circuitry on the previous circuitry converts capacitance measurements (~5pF to 250pF) into a frequency signal (with a range of roughly 640 to 890Hz) at 5V. The new interface PCB down-converts the 5V signal to a 3.3V signal (through a voltage divider) before the input pin to the MSP430FR5949.

There are two methods of using the Rain Gauge currently; 60minute report mode and direct measurement mode.

In the 60 minute report mode, the number of counts in one second are stored into a 60 second buffer. On the minute mark (as alarmed by the RTC), the counts are converted to volumes (through metadata values input from the user)and the mean, standard deviation, min and maximum volumes are stored into a buffer that is 60 minutes deep. The 60 minute buffer is a circular buffer, so at minute 61 it is overwriting minute 1, and therefore always saves the last 60 minutes of data.

In direct measurement mode, the user can request either the current volume or total number of counts since the last request. Once requested, the sensor completes its current second of measurement, and computes the volume (if requested) and displays the calculated volume or sum of counts, and the number of seconds elapsed since the last measurement. This mode is useful for making an instantaneous measurement and for calibration.

# Operating Procedures

## Electrical Characteristics

Table 1 shows the electrical characteristics of the rain gauge.

Table 1. System Electrical Characteristics

|  |  |
| --- | --- |
| Voltage Input | 2.5 – 24VDC |
| Current Draw @12VDC (RS232 Port Connected) | 4.35mA |
| Current Max @12VDC (RS232 Port Connected) | 5.63mA |
| Current Draw @12VDC (RS232 Port Disconnected) | 1.21mA |
| Current Draw @12VDC (RS232 Port Disconnected) | 2.30mA |

## Physical and Serial Connections

Figure 1 diagrams the connection between the user and the Rain Gauge.

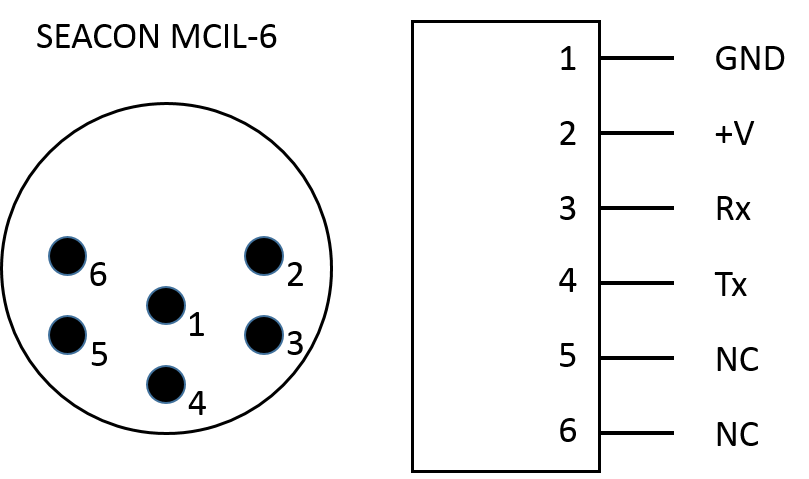


Figure 1. Rain Gauge Power/Signal Connector

The serial connection parameters for connecting with the rain gauge are documented in Table 2.

Table 2. RS-232 Settings

|  |  |
| --- | --- |
| Baud Rate | 9600 |
| Bits | 8 |
| Parity | None |
| Stop Bit | 1 |
| Transmit | CR & LF |

## User Interface

When connected to power, the Rain Gauge automatically begins sampling. As mentioned in Section 2, there are two modes of operation: 60 minute reporting mode and direct measurement mode. Data for the 60 minute reporting mode is available after the first minute of operation (will read all zero until then). Data for the direct measurement mode is available after the first second of operation.

### Data Retrieval – Direct Measurement Mode

There are two types of direct measurement data that can be requested from the Rain Gauge: Volume and Sum of Counts. The volume measurement converts the raw data stored in the direct measurement buffer and converts it to mL using the metadata stored in memory. The Sum of Counts measurement returns the sum of all counts from the capacitance sensor since the last request or power-on of the device.

To request the Volume Measurement, transmit a “D” character. The response will be:

**@@@x.ymL,z\r\n**  
where:

@@@ is the three @ (0x40) character header  
x .y is the volume with 2 decimal place precision (4 real numbers)  
mL is the units of milliliters  
z is the number of seconds elapsed  
\r is a carriage return (0x0D)  
\n is a line feed (0x0A)

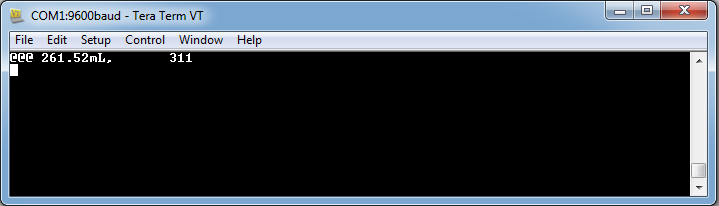


Figure 2. Sensor volume response to "D" command

To request the Sum of Counts Measurement, transmit a “d” character. The response will be:

**@@@x,y\r\n**  
where:

@@@ is the three @ (0x40) character header  
 x is the sum of counts since the last measurement  
 y is the number of seconds elapsed  
 \r is a carriage return (0x0D)  
 \n is a line feed (0x0A)



Figure 3. Sensor sum of counts response to "d" command

### Data Retrieval – 60 Minute Report

The Rain Gauge sensor stores 60 minutes’ worth of rain data in memory. The data stored is the statistical measurements made on 60 seconds worth of data, and include: Mean Volume, Standard Deviation, Minimum Volume and Max Volume.

To request the 60 minute report, transmit an “R” or ‘r’. The response will have the format of:

@@@YYYYMMDD,HH:MM,Mean,STD,Min,Max\r\n  
where:  
 YYYY is 4-digit Year  
 MM is 2-digit Month  
 DD is 2-digit Day  
 HH is 2-digit Hour on 24-Hour Clock  
 MM is 2-digit Minute  
 Mean is the mean volume over that minutes 60 seconds  
 STD is the standard deviation of the volumes over that minutes 60 seconds  
 Min is the minimum volume over that minutes 60 seconds  
 Max is the maximum volume over that minutes 60 seconds

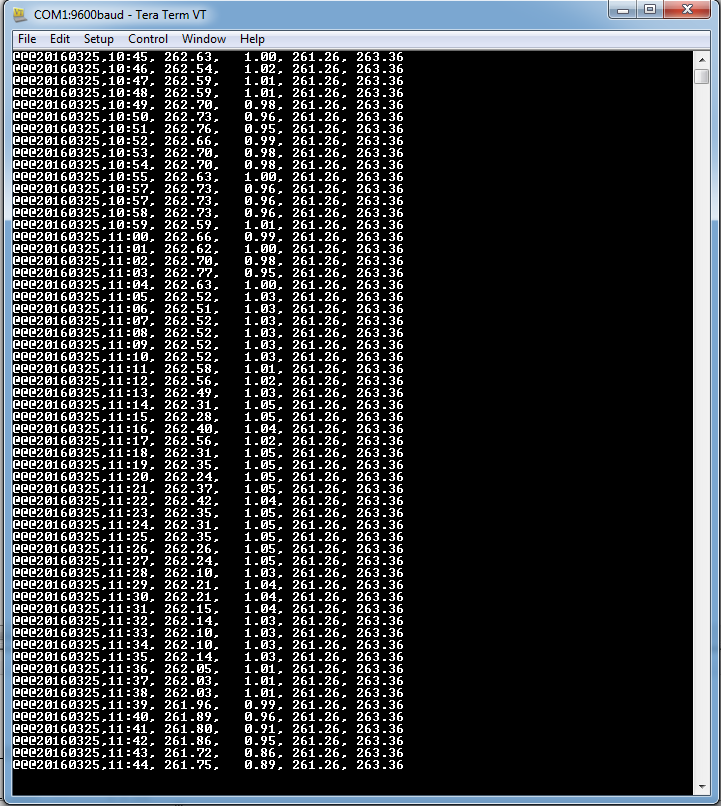


Figure 4. 60 Minute Report

### Input Console

To enter the console mode, press “Ctrl-C” (0x18) three (3) times within 10 seconds. A right bracket (“>”) will be transmitted over the UART.

The typical structure for entering a new value is:

**CC XXXXX\r\n**  
Where:  
 CC is a command  
 Space (0x20) between the command and the value  
 XXXXX is the value  
 \r is a carriage return (0x0D)  
 \n is a line feed (0x0A)

Command codes can be found in Appendix A: Console Commands.

**EXAMPLE: Slope Input**

To input the slope of 1000000000000, the user would enter:

**A0 1000000000000**  
or  
 **A0 1.0e12**

**EXAMPLE: Intercept Input**

To input the slope of 1000000000000, the user would enter:

**A1 -500**  
or  
 **A1 -5.0e2**

**EXAMPLE: DateTime Input** **March 26, 2016, 11:47:23AM**

If the user wants to set the clock to March 26, 2016, 11:47:23AM, they would enter:  
 **T 20160325114723**

**EXAMPLE: DateTime Input** **Dec 3, 2016, 4:09:00PM**

If the user wants to set the clock to March 26, 2016, 11:47:23AM, they would enter:  
 **T 20161203160900**

**EXAMPLE: Serial Number Input**

If the user wants to set the serial number to RG2016-0001, they would enter: **S RG2016-0001**

# Appendix A: Console Commands

List of available console commands:

|  |  |
| --- | --- |
| **Command** | **Function** |
| **A0** or **a0** | Input **SLOPE** value for volume conversion. Accepts floating point value input. |
| **A1** or **A1** | Input **INTERCEPT** value for volume conversion. Accepts floating point value input. |
| **T** or **t** | Input RTC **TIME**. Accepts decimal number in YYYYMMDDHHMMSS format |
| **S** or **s** | Input **SERIAL NUMBER.** Accepts string up to 16 characters long. |
| **Ctrl-X** or **(0x1B)** | **EXIT** console mode. No Input. |
| **?** | Query device **METADATA.** No input. |
|  |  |

# Appendix B: List of Acronyms

NOAA – National Oceanic and Atmospheric Administration  
PMEL – Pacific Marine Environmental Laboratory  
UART – Universal Asynchronous Receiver/Transmitter  
SPI – Serial Peripheral Interface  
RTC – Real-Time Clock  
Hz – Hertz  
mA – Milli-amps  
uA – Micro-amps  
V – Volts  
VDC – Volts (Direct Current)  
Tx – Transmit  
Rx – Receive  
GND – Ground  
PWM – Pulse-width Modulated  
FRAM – Ferromagnetic Random Access Memory  
mL – milli-liter  
PCB – Printed Circuit Board

# Appendix C: Mathematical References

## Statistical Data Pseudo-Code:

Mean:

**for (i = 0; i < length of data set; i++)**

**sum += data[i]**

**MeanValue = sum/length;**

**Max:**

**for (i = 0; i < length of data set; i++) {**

**if(data[i] > DataMax) {**

**DataMax = data[i] }}**

Min:

**for (i = 0; i < length of data set; i++) {**

**if(data[i] < DataMax) {**

**DataMax = data[i] }}**

STD:

**for (i = 0; i < length of data set; i++) {**

**Diff = data[i] – MeanValue  
 Diff \*= Diff  
 Variance += Diff }**

**Variance /= length**

**STD = sqrt(Variance)**

## 

## Volume Calculations formula:

For the Rain gauge, the volume is calculated using the formula:

Where:  
 A0 is the calibrated system SLOPE  
 A1 is the calibrated system INTERCEPT

dm is a calculated value that is given by:

Where:  
 n is the Sum of sensor pulses (counts)  
 s is the total number of elapsed seconds since n=0  
 60 scales value to one minute