APPLICATION NOTE

App. Note Code: 2MI-U
Rev. 1

Ott Parsivel Disdrometer Present Weather Sensor

CAMPBELL SCIENTIFIC®
WHEN MEASUREMENTS MATTER

Ott Parsivel Disdrometer Present Weather Sensor

This application note describes using the Ott parsivel Disdrometer Present Weather Sensor with Campbell Scientific dataloggers. Specifically the document includes installation information and programming.

General Information

The Parsivel Disdrometer measures precipitation and can distinguish between

- drizzle
- drizzle with rain
- drizzle with snow
- rain
- snow
- snow grains
- freezing rain
- hail

There are two interface options available for the Parsivel. One is SDI-12, and the other is serial RS-485. SDI-12 can be used to interrogate the sensor for pre-defined messages. The Parsivel needs to be reconfigured for this mode.

The Parsivel can be used out of the box with the RS-485 bus. In this case, the whole measurement set of data can be acquired and parsed according to user needs.

Hardware

Setup

Hyperterm is used to make configuration changes to the Parsivel.

Do not attempt to set the instrument into RS-485 bus mode. If you do this, an address will be required to talk to the device. Because the address is not known, the factory will need to fix the sensor.

If SDI-12 is used, Hyperterm is required and the message sent is CS/S/E 1.

There are no adjustments needed for two wire RS-485 mode.

Installation

Use the 18663 null modem cable to connect the datalogger's RS-232 port to the MD485's RS-232 port. The Parisivel connects to the 9721 cable as follows:

Parsivel terminal	9721 cable
1 pwr gnd	black
2 pwr gnd	red
3 RS-485 B	blue
4 RS-485 A	yellow
5 SDI-12	white
6 digital pulse	green
7 gnd	shield

815 W. 1800 N., Logan, UT 84321-1784 (435) 753-2342

For RS-485 mode, tThe sensor attaches to the MD485 as follows:

MD485	Sensor
gnd	clear (shield)
RS-485 A	yellow
RS-485 B	blue
12V	red
gnd	black

The Parsivel attaches to a crossarm using a series of pipe reducers to go from 1 ½ inches to 1 inch. A suggestion that works is

- 1 ½ inch by 6 inch nipple
- 1 ½ inch coupling
- 1 ½ inch to 1 inch bushing
- 1 inch by 6 inch nipple

A six foot crossarm is used with an additional CM210 crossarm to pole mounting kit to give additional strength to the mount for the weight of the sensor. This configuration would be used to attach the sensor to a tower.

Programming

The program provided acquires the entire data set and is parsed for the particle speed, and size outputs.

```
'CR1000 Series Datalogger

'date:

'program author:

'No output is written in this version

'sensor wiring

'MD-485 via null modem cable      logger
'rs-232 port                  rs-232 port

'Parsivel terminal             9721 cable
'1 pwr gnd                     bk
'2 pwr                         rd
'3 rs-485 b                    bl
'4 rs-485 a                    yl
'5 sdi-12                      wh
'6 digital pulse                 gn
'7 gnd                         sh
```


815 W. 1800 N., Logan, UT 84321-1784 (435) 753-2342

```
'Parsivel          md-485
'bl                rs-485 B
'yl                rs-485 A
'shield            G

'Parsivel          Power
'rd                +12v
'bk                G

'Declare Public Variables

'the raw data variable
Public csp as string * 5000,csppoll as string,test90split(32),test91split(30),filter90,filter91

'the first split to remove the size and speed variables
public test90 as string * 226
public test91 as string * 31

'alias instruction s to define the classes

'size classes
'the size classes only have 30 classes due to the sensors first two classes being
'outside the range of the sensor. -9.999 is the no measurement output for the
'size measurements and the logger outputs -10.  -9.999 is always output for the
'first two classes.

alias test90split(1)=mm062
alias test90split(2)=mm187
alias test90split(3)=mm312
alias test90split(4)=mm437
alias test90split(5)=mm562
alias test90split(6)=mm687
alias test90split(7)=mm812
alias test90split(8)=mm937
alias test90split(9)=mm1062
alias test90split(10)=mm1187
alias test90split(11)=mm1375
alias test90split(12)=mm1625
alias test90split(13)=mm1875
alias test90split(14)=mm2125
alias test90split(15)=mm2375
alias test90split(16)=mm2750
alias test90split(17)=mm3250
alias test90split(18)=mm3750
alias test90split(19)=mm4250
alias test90split(20)=mm4750
alias test90split(21)=mm5500
alias test90split(22)=mm6500
alias test90split(23)=mm7500
alias test90split(24)=mm8500
alias test90split(25)=mm9500
alias test90split(26)=mm11000
alias test90split(27)=mm13000
alias test90split(28)=mm15000
alias test90split(29)=mm17000
```


815 W. 1800 N., Logan, UT 84321-1784 (435) 753-2342

```
alias test90split(30)=mm19000
alias test90split(31)=mm21500
alias test90split(32)=mm24500

'speed classes
'the speed classes only have 30 classes due to the sensors first two classes being
'outside the range of the sensor.  Since the no measurement output is nothing,
'the first two "outside the range" measurements don't get registered, and the logger
'moves all values up two positions.  So, data actually starts with the speed on the
'third element of the speed output.

alias test91split(1)=ms250
alias test91split(2)=ms350
alias test91split(3)=ms450
alias test91split(4)=ms550
alias test91split(5)=ms650
alias test91split(6)=ms750
alias test91split(7)=ms850
alias test91split(8)=ms950
alias test91split(9)=ms1100
alias test91split(10)=ms1300
alias test91split(11)=ms1500
alias test91split(12)=ms1700
alias test91split(13)=ms1900
alias test91split(14)=ms2200
alias test91split(15)=ms2600
alias test91split(16)=ms3000
alias test91split(17)=ms3400
alias test91split(18)=ms3800
alias test91split(19)=ms4400
alias test91split(20)=ms5200
alias test91split(21)=ms6000
alias test91split(22)=ms6800
alias test91split(23)=ms7600
alias test91split(24)=ms8800
alias test91split(25)=ms10400
alias test91split(26)=ms12000
alias test91split(27)=ms13600
alias test91split(28)=ms15200
alias test91split(29)=ms17600
alias test91split(30)=ms20800

'Define Data Tables

'Define Subroutines

'Main Program

BeginProg
  'filters that determine where the data starts in the raw record
  filter90 = chr(57)+chr(48)+chr(58)        '90:
  filter91 = chr(57)+chr(49)+chr(58)        '91:

  'polling command cs/pa
```


815 W. 1800 N., Logan, UT 84321-1784 (435) 753-2342

```
  csppoll=CHR(67)+CHR(83)+CHR(47)+CHR(80)+CHR(65)+CHR(13)+CHR(10)  'CS/
PA(cr)(lf)

  Scan (15,Sec,0,0)
      'opens the serial port and sets the communication parameters
      SerialOpen (COMRS232,-19200,0,0,10000)

      'the polling command is sent
      SerialOut (COMRS232,csppoll,"",0,50)

      'raw data is received
      SerialIn (csp,COMRS232,50,chr(13)+chr(10)+chr (03),5000)

      'the first split to get the size classes
      'test 90 - size 32 classes
      SplitStr (test90,csp,filter90,1,4)

      'the second split to get individual size measurements.  -10 equals no data
      SplitStr (test90split(),test90,chr(59),32,0)

      'the first split to get the speed classes

      'test 91 - speed 32 classes
      splitstr (test91,csp,filter91,1,4)

      'the second split to get the individual speed measurements.  NAN equals no data
      SplitStr (test91split(),test91,chr(59),32,0)

      'clears the buffers pprior to the next measurement
      SerialFlush (COMRS232)

  NextScan
EndProg
```