Interfacing the eosFD to a Campbell Scientific CR1000/CR6 Data Logger

Introduction

While the Eosense eosFD CO_2 flux chamber has internal data storage, users may wish to connect it to an external data logger to synchronize readings with those from other sensors. The Campbell Scientific CR1000 and CR6 are popular data loggers that provide a plethora of options for connecting and recording data from multiple sensors. This application note describes both analog and serial connection methods to help you decide which method is right for your application.

Required Components

- eosFD flux chamber
- Campbell Scientific CR1000/CR6 data logger
- Short Cut software or CRBasic Editor
- Power and Data Cable (SSC/SLC)
- Power Connector
- Ground/Serial Breakout Connector

Differential Voltage Mode (CR1000)

The eosFD analog outputs can connect to the CR1000 differential inputs. Plug the included 3-pin ground connector into the Power and Data Cable. This supplies a separate ground reference for each analog output. The analog/ground pairs connect to high/low (H/L) differential input pairs on the CR1000 as shown in **Figure 1**. Plug the power connecter into the Power and Data Cable, then connect the red and black wires either to the 12V output of the CR1000 or to a separate 12V power source.

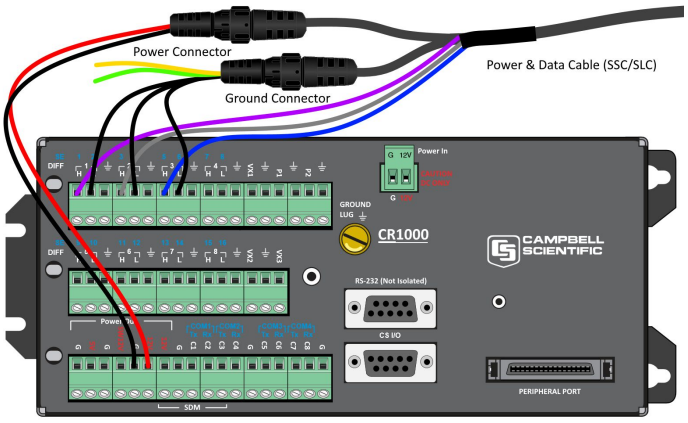

Figure 1: Connecting the eosFD to a CR1000 in differential mode (printable image). Note that the yellow and green serial wires are unused in this configuration.

This setup can be configured using the Campbell Scientific Short Cut program. See the Campbell Scientific website for a primer on using Short Cut: *Programming with Short Cut (QuickStart Part 2)*. In the sensor section, select “Differential Voltage” and add three sensors. Adjust the properties for each sensor as per **Table 1**. The eosFD will report CO_2 flux, CO_2 sensor temperature, and concentration in the eosFD soil cavity by default.

Analog Property	Purple Wire	Grey Wire	Blue Wire
Measurement Result (name)	CO2_Flux	Sens_Temp	Soil_Conc
Measurement Result (units)	µmol/m²/s	Degrees C	ppm
Range of Sensor Voltage	0 to 5000 mV	0 to 5000 mV	0 to 5000 mV
Multiplier	0.004	0.012	1.0
Offset	-10	-20	0

Table 1: Analog properties of the default eosFD outputs. Note that signal wires can be connected selectively. For example, if only CO_2 flux is desired, only the purple wire and ground need be connected. This leaves terminals free for other connections.

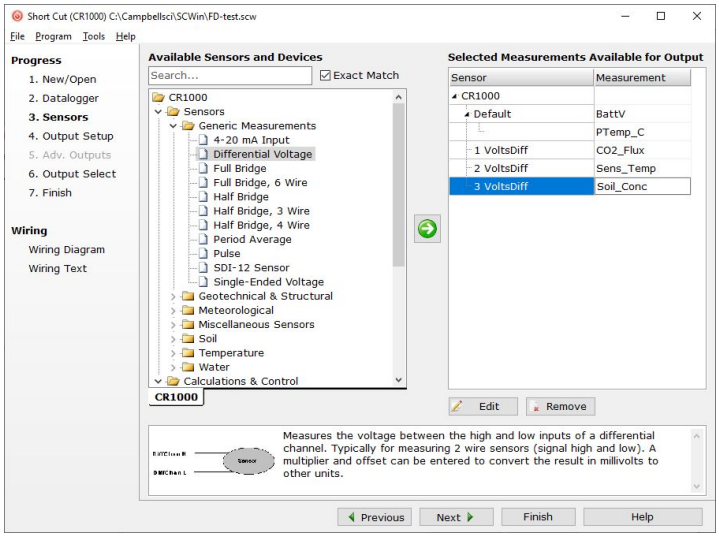

Figure 2: Short Cut Differential Voltage set-up for all three eosFD analog outputs.

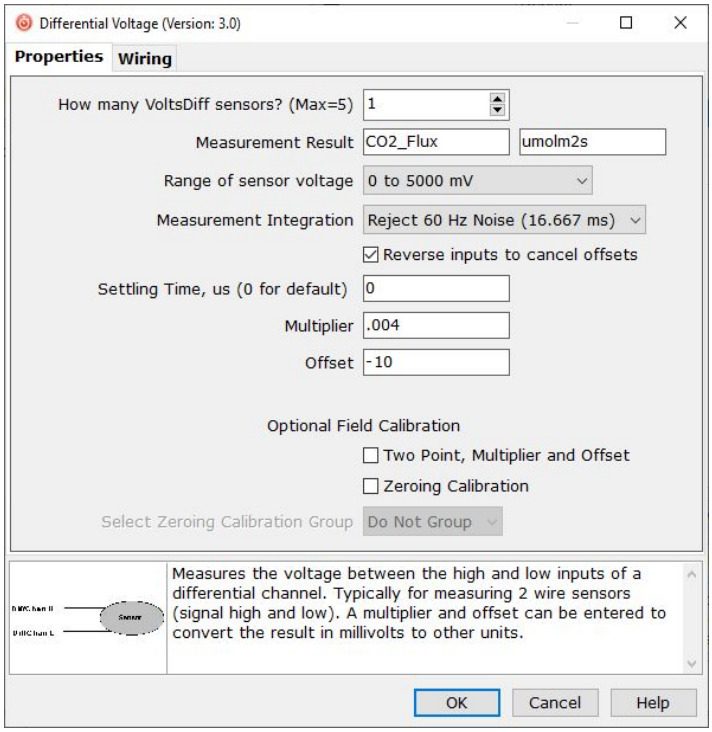

Figure 3: Short Cut properties box showing settings for analog CO2_Flux measurements.

Single-Ended Voltage Mode (CR1000)

If differential terminals are unavailable, the eosFD can use the single-ended mode. However, eosFD single-ended measurements are less accurate, and should be avoided unless absolutely necessary. Because the eosFD draws large but short-duration current pulses to power its NDIR sensor, pump, and valves, a varying amount of voltage is dropped across the ground terminal of the cable (an effect exacerbated by longer cables). This time-varying noise signal can be diminished by using an isolated power supply that is separate from the CR1000's supply without a common ground connection (**Figure 4**). The configuration in Short Cut uses “Single-Ended Voltage” as the sensor type, but all other settings are the same as for differential mode (**Table 1**).

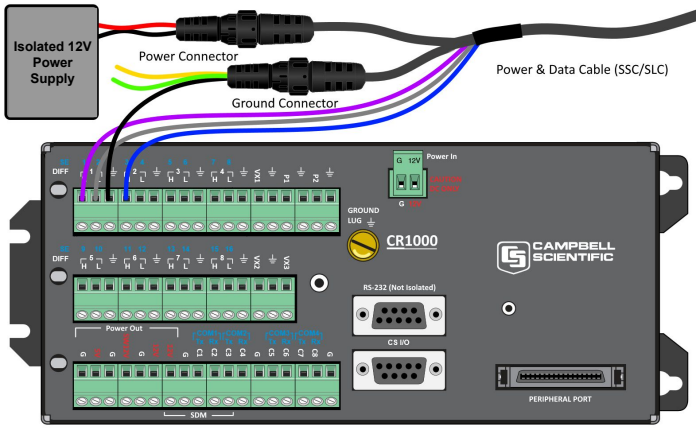

Figure 4: Connecting an eosFD to a CR1000 in single-ended mode (printable image). Note that the yellow and green serial wires are unused in this configuration.

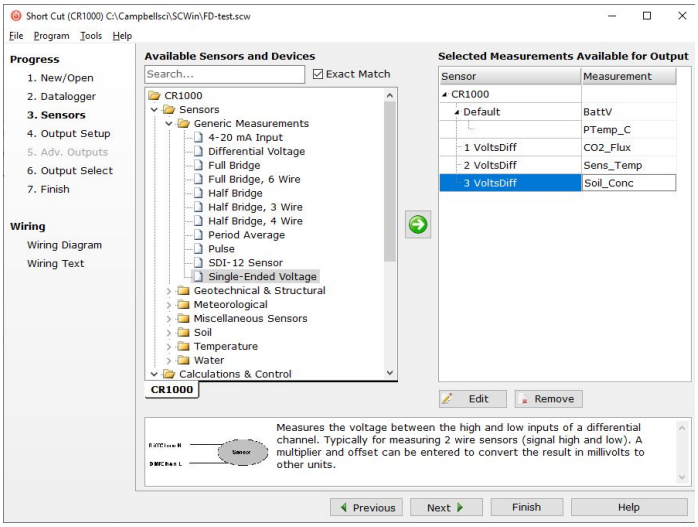

Figure 5: Short Cut Single Ended set-up for all three eosFD outputs

Voltage Modes in the CR6

The CR6 datalogger uses the same modes (differential and single ended) as the CR1000. The main difference between the dataloggers for this application is the number of terminals available for voltage logging. The CR1000 has eight differential inputs, while the CR6 universal terminals can be configured as up to six differential inputs.

Other Supported Loggers

The eosFD is compatible with most Campbell Scientific data loggers, with slight modifications to the programming and the input locations and style. To the right is a list of data loggers compatible with differential and single ended measurement modes.

CR6 CR300
CR800 CR1000
CR1000X CR3000

© Eosense Inc.

Serial Inputs

While the analog outputs are the simplest to use, it is preferable in most instances to use the eosFD serial output for maximum resolution and noise immunity. Note that while Short Cut is the recommended tool for creating a data logger program for analog inputs, the CRBasic Editor is needed for monitoring serial data.

If you have an older eosFD, you may need to update it with the latest Eosense firmware to activate the serial interface (contact support@eosense.com for assistance).

The CR1000 allows up to four serial inputs (labelled COM1 to COM4) on its control I/O ports, in addition to other inputs. The newer CR6 allows two serial inputs on its control I/O ports (an additional RS-232 serial input is available on the RJ11 connector labelled CPI, but this requires a special adapter cable from Campbell Scientific). If using an eosFD with serial interfacing to an external data logger, obtain a serial breakout adapter from Eosense. Depending on which port(s) you plan to use, connect each serial breakout cable as shown in **Tables 2** and **3** below and in **Figures 6** and **7**.

CR6 Data Serial Connection

Wire	Signal	ComC1	ComC2
Yellow	Tx (to eosFD)	C1	C3
Green	Rx (from eosFD)	C2	C4
Black	Ground	G	G

Table 2: Serial connections to a Campbell Scientific CR6 data logger.

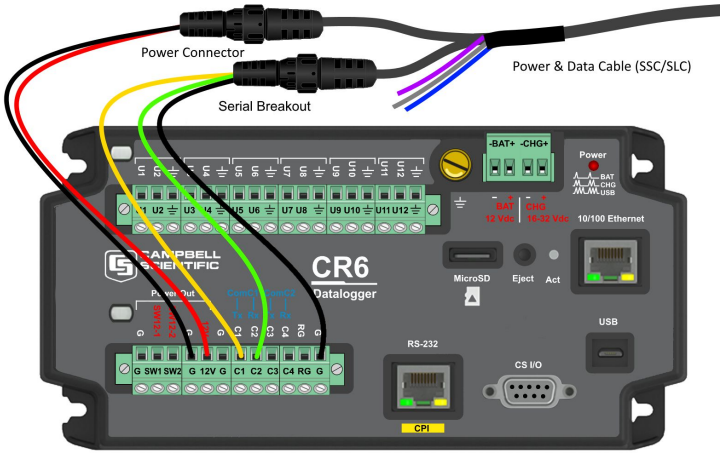

Figure 6: Serial data connection of the eosFD to a CR6 on ComC1 (printable image). Note that the purple, grey, and blue analog wires are unused in this configuration.

For the CR6, connect the green wire of the eosFD's serial breakout cable to the Rx input of the CR6 (labelled C1 for the ComC1 port), and the black wire to a serial ground (usually labelled 'G') as shown in **Figure 6**. The transmit wire is not used in this application, but the yellow wire can be connected to the Tx input (labelled C2 for the ComC1 port). The eosFD can either be powered from the data logger 12V supply or via a separate supply. In this example, ComC1 is used, but with a small change to the program, any of the control I/O COM Ports can be used.

CR1000 Serial Connection

Wire	Signal	COM1	COM2	COM3	COM4
Yellow	Tx (to eosFD)	C1	C3	C5	C7
Green	Rx (from eosFD)	C2	C4	C6	C8
Black	Ground	G	G	G	G

Table 3: Serial connections to a Campbell Scientific CR1000 data logger.

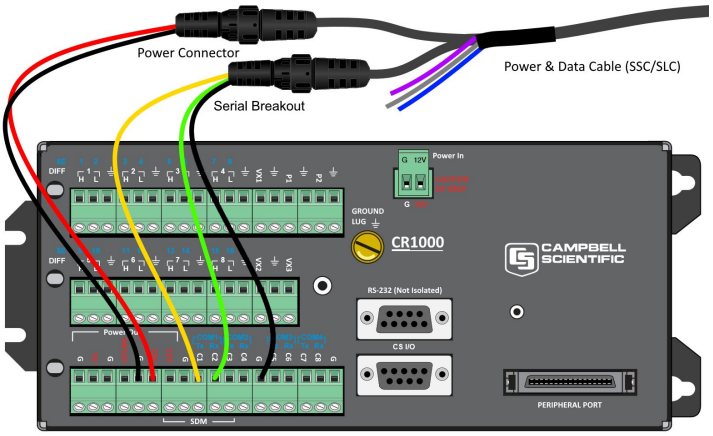

Figure 7: Serial data connection of the eosFD to a CR1000 on COM1 (printable image). Note that the purple, grey, and blue analog wires are unused in this configuration.

For the CR1000, connect the green wire of the eosFD's serial breakout adapter to the Rx input of the CR1000 (labelled C1 for the COM1 port), and the black wire to a serial ground (usually labelled 'G') as shown in **Figure 7**. The transmit wire is not used in this application, but the yellow wire can be connected to the Tx input (labelled C2 for the COM1 port). The eosFD can either be powered from the data logger 12V supply or via a separate supply. In this example, COM1 is used, but with a small change to the program, any of the four control I/O COM Ports can be used.

© Eosense Inc.

Sample Data Logger Program

A demonstration program is shown in **Table 4** and is available as a download (CR6, CR1000) from the Eosense website. The text shows a simple CRBasic program which records time-stamped datalogger battery voltage and eosFD serial data.

Lines **042** and **064** contain references to **ComC1**. These can be changed to **ComC2** on the CR6 or **Com1**, **Com2**, **Com3** or **Com4** on the CR1000 to make use of the other serial ports. The eosFD outputs the last reading from its serial port approximately every five seconds regardless of what the measurement interval is set to. After each measurement cycle, the serial message is updated to reflect the most recent reading. The date and time in the serial message is the time that the measurement was started, based on the eosFD's internal clock. When each new serial message arrives, the CRBasic program updates the measurement values (**Scan( )** should have a frequency of less than 5 seconds). Calling the data table will store these most recent values.

Averaging can be used to reduce the frequency of the eosFD data to match that of other sensors that are measured less frequently. While the demonstration program was developed specifically for the CR6 and CR1000, the techniques are applicable to other data logger models like the CR800 and CR3000 with minor changes to the program.

Online Resources

The wiring diagrams and data logger programs used in this document are also available online!

Table 4 (right): Source code for a sample eosFD monitoring program (FD_Serial_Demo.CR6)

```
001     'CRBasic Logger eosFD Serial Demo created by Eosense
002     'Declare Variables and Units
003     Public BattV
004     Public SerialInput As String * 100
005     Public SerialFields(13) As String * 16
006     Public FD_Month as LONG
007     Public FD_Day as LONG
008     Public FD_Year as LONG
009     Public FD_Hour as LONG
010     Public FD_Minute as LONG
011     Public FD_Second as LONG
012     Public FD_Flux as FLOAT
013     Public FD_Temp as FLOAT
014     Public FD_Soil as FLOAT
015     Public FD_Soil_STD as FLOAT
016     Public FD_Atm as FLOAT
017     Public FD_Atm_STD as FLOAT
018     Units BattV=Volts
019     Units FD_Soil=ppm CO2
020     Units FD_Atm=ppm CO2
021     Units FD_Temp=Deg C
022
023     'Define Data Tables
024     DataTable(Table1,True,-1)
025             DataInterval(0,1,Min,10)
026             Sample(1,BattV,FP2)
027             Sample(1,FD_Temp,FLOAT)
028             Sample(1,FD_Flux,FLOAT)
029             Sample(1,FD_Soil,FLOAT)
030             Sample(1,FD_Atm,FLOAT)
031             Sample(1,FD_Month,LONG)
032             Sample(1,FD_Day,LONG)
033             Sample(1,FD_Year,LONG)
034             Sample(1,FD_Hour,LONG)
035             Sample(1,FD_Minute,LONG)
036             Sample(1,FD_Second,LONG)
037     EndTable
038
039     'Main Program
040     BeginProg
041             'Initialize Serial Port (CR1000: Com1... Com4  CR6: ComC1/ComC2)
042             SerialOpen(ComC1,115200,0,0,100)
043
044             'Initialize variables.
045             FD_Month = -2147483647
046             FD_Day = -2147483647
047             FD_Year = -2147483647
048             FD_Hour = -2147483647
049             FD_Minute = -2147483647
050             FD_Second = -2147483647
051             FD_Flux = NAN
052             FD_Temp = NAN
053             FD_Soil = NAN
054             FD_Soil_STD = NAN
055             FD_Atm = NAN
056             FD_Atm_STD = NAN
057
058             'Main Scan
059             Scan(2,Sec,1,0)
060                     'Default Datalogger Battery Voltage 'BattV'
061                     Battery(BattV)
062                     'Read eosFD Serial Data (for CR1000 use
063                     'Com1/Com2/Com3/Com4, for CR6 use ComC1/ComC2)
064                     SerialIn(SerialInput,ComC1,6,&h0D,100)
065                     'If new data is available, parse and convert
066                     if InStr(1,SerialInput,",",2) then
067                             'Separate input into fields
068                             SplitStr(SerialFields(),SerialInput,",",13,0)
069                             'Store time/date as integers
070                             FD_Month=SerialFields(2)
071                             FD_Day=SerialFields(3)
072                             FD_Year=SerialFields(4)
073                             FD_Day=ABS(FD_Day)
074                             FD_Year=ABS(FD_Year)+2000
075                             FD_Hour=SerialFields(5)
076                             FD_Minute=SerialFields(6)
077                             FD_Second=SerialFields(7)
078                             'Store sensor data as floating point numbers
079                             FD_Flux=SerialFields(8)
080                             FD_Temp=SerialFields(9)
081                             FD_Soil=SerialFields(10)
082                             FD_Soil_STD=SerialFields(11)
083                             FD_Atm=SerialFields(12)
084                             FD_Atm_STD=SerialFields(13)
085                     endif
086                     'Call Data Table and Store Data
087                     CallTable Table1
088             NextScan
089     EndProg
```


© Eosense Inc.