**Abbreviations**

CR1000X CRX

Device Configuration Utility DCU

MetSEN300 MS3

R3-50 Anemometer R350

SDM-SIO1A Serial I/O Monitor SDM

SDM-SIO4A Serial I/O Monitor SDM4

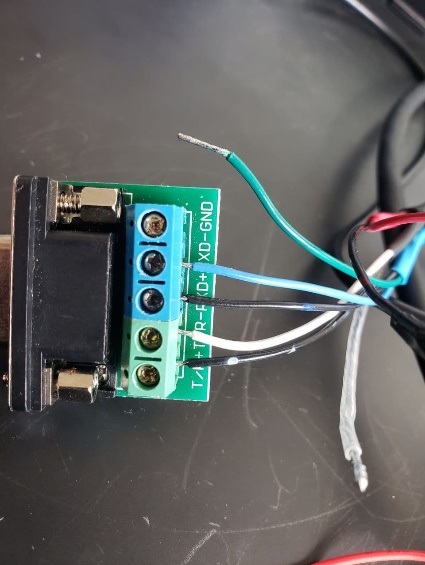
PC400 Datalogger Software PC

LORD 3DM-GX5-35 IMU

Hygrovue 10 Temp & RH Probe HYG10

EXO2 Sonde, YSI Instruments EXO

**March 25th, 2021 – Connecting HS50 to computer and PC400**

Green from blue Ground Serial

Blue from blue RXD- ‘

Black from blue RDX+ ‘

White from black T/R- ‘

Black from black T/R+ ‘

Red from red 12v power in Power Supply

Black from red Ground ‘

Clear is the chassis ground, which we did not connect to anything

As the serial connecting doesn’t have power, the HS50 must be connected directly to a 12v power supply.

**April 1st, 2021 – Connecting MetSEN300 to the CR1000X Datalogger, uploading program, testing equipment**

Program: *2021.4.1 CR1000X METSens*

Power the CRX with 12v (the power supply supplies 13.8v and that’s okay). Connect to it via USB and the programs *Device Configuration Utility* (DCU, must right click and select “Run as administrator”) or *PC400*.

Wiring the MS3 using the SDI-12 interface

Red 12v

Black Ground

Clear Analog ground (The “classic” ground symbol, not the ones labeled G)

Green Data, into whatever C port is specified in the program. MUST be C1, C3, C5, or C7.

*SDI12Recorder (MSData(1),****C5****,0,"MC1!",1.0,0,9)*

When using DCU

* Select the COM port showing the CRX (there should only be one showing the CRX). 9600 baud rate.
* Send programs to the CRX in the Logger Control window. Bottom left “Send Program”.
* View live data from the CRX in the Data Monitor window. Select table on left to view the data. If the data is showing “NAN” either the green data wire is in the incorrect C port, or the program is trying for the wrong port.

To view the saved data files

Open the program PC400, connect to CRX, and go to the Collect Data tab. Start Data collection which will save a file onto the computer (by start data collection it means start collecting the data on the CRX, not start collecting data on the CRX). Select the file you desire and View. This will open the View program from PC400, which is an easy way to verify the information is saved correctly in the file and to view quick graphs of the information.

**April 8th, 2021 – HS-50 Sonic anemometer onto the CR1000X, through SDM-SIO1A Serial I/O Module**

Program: 2021.4.8 CR1000X HS50 through SDM-SIO1A

Wiring the R3-50 to SDM

Green from blue GND

Blue from blue CTS/RD-

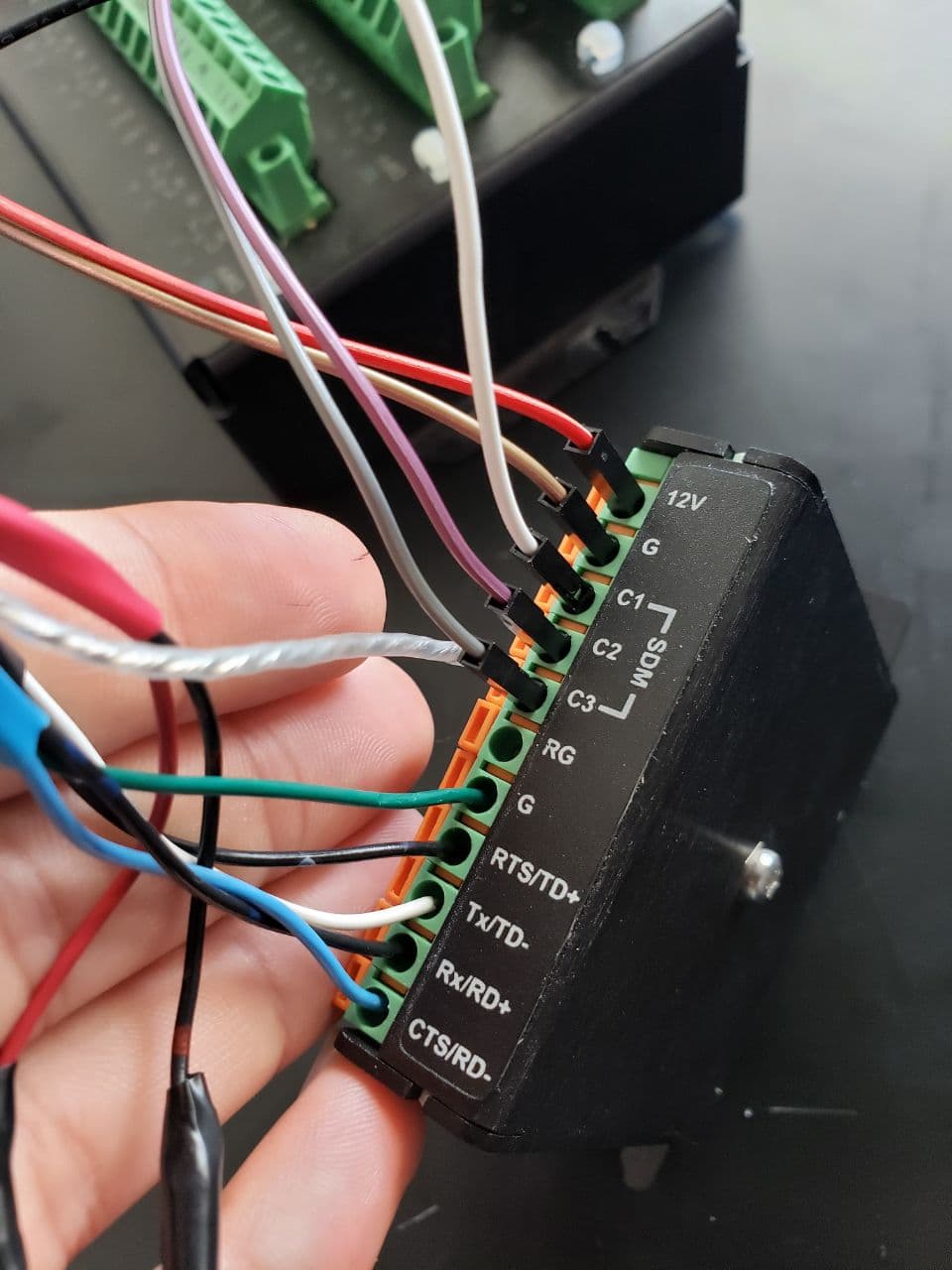
Black from blue Rx/RD+

White from black Tx/TD-

Black from black RTS/TD+

Clear is the chassis ground, which we did not connect to anything

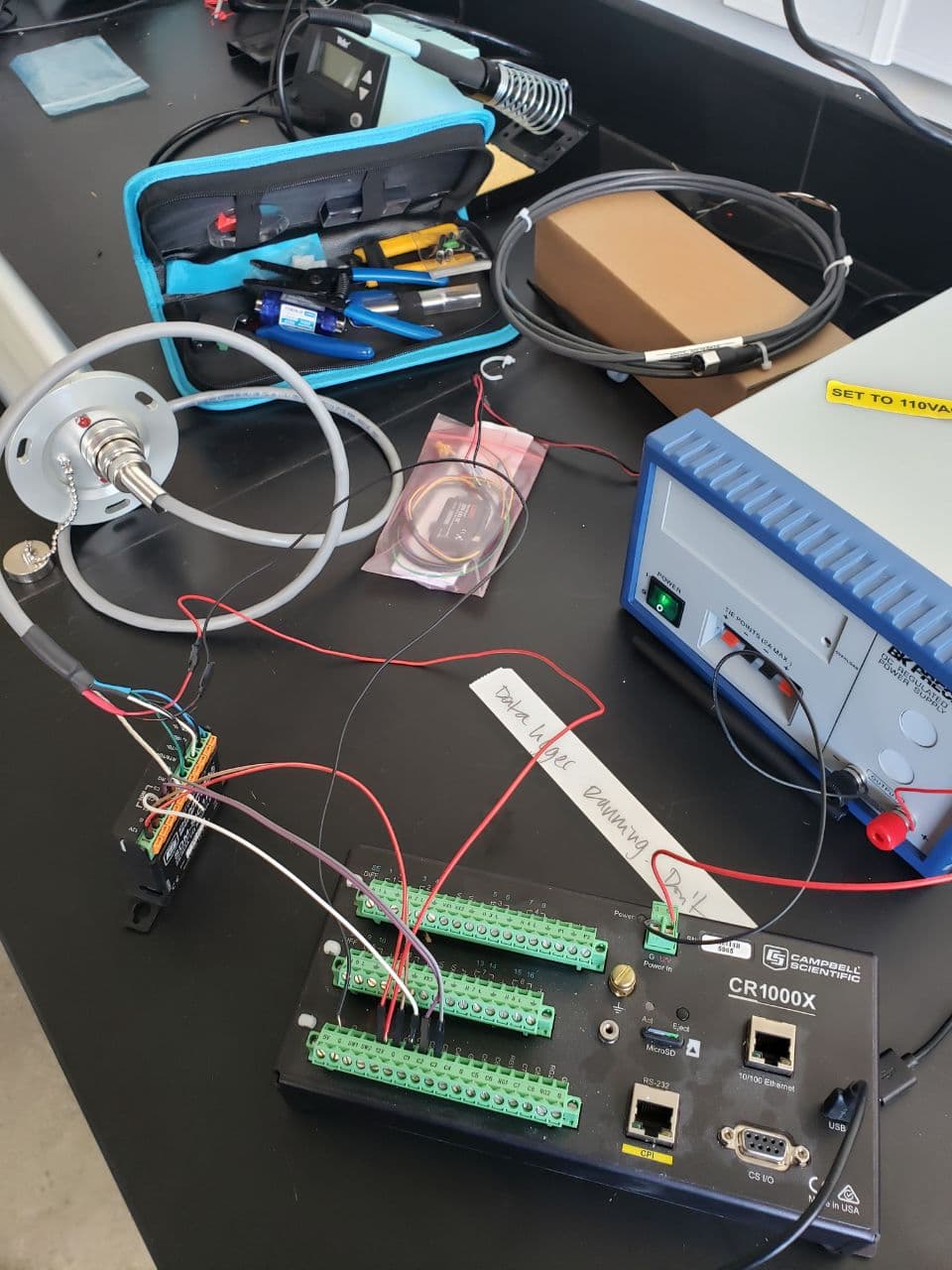
Connect the 12v and G from the HS50 directly to the CRX 12v and G. There will be two 12v power cables from the 12v on the CRX, one for the HS50 and one for the SDM. The actual wire colors are not important at this juncture (as you can see in both photos) since this is simply for testing.



From SDM to CRX

HS-50

12v 12v

 G G

C1 C1

C2 C2

C3 C3

It’s as simple as it looks, connecting the SDM to CRX. Power and numbered C’s, as you should be good!

CR1000X

**April 15th, 2021 – Connecting the LORD MicroStrain 3DM-GX5 IMU**

ONLY USE ONE OF THE POWER CABLES, IN THIS CASE #6 FOR 12V

PPS output is a 1Hz output signal from the unit based on satellite time, and is supposedly a very accurate

1 USB D- Black

2 USB D+ Brown

3 4V-5.5V Red

4 RS232 RxD Orange

5 RS232 TxD Yellow

6 5.2V-36V Green

7 PPS output Blue

8 GND Purple

9 - Gray

* Tried to connect the IMU to the computer though the serial cable and MIP Monitor, to no success.
* Then tried connecting the IMU to the datalogger to no success with the original IMU code commented out in the HS50.
* Put the code together for the MS3, HS50, and IMU and got it to compile. Connected the HS50 via the same SDM and added the MS3 and got it at least display the data out on the computer. IMU nothing, since not connected to anything.
* Tried using the SMD4 to connect the HS50 and IMU together, along with adding the MS3 to the mix. Connected everything and it didn’t work.

Plans for next steps:

* Review the combined code to see if it is okay and seems like it should work
* Disconnect all wires, reconnect while reviewing the wiring
* Ensure the SMD4 is working as intended, maybe with only the HS50 first (datasheets/manuals on them?)

**April 22nd, 2021 – Connecting the IMU to CRX viz COM7 and COM8**

Program: sttr - original from old sensor suite

Today is a new day! Re-wiring the whole thing as it became sort of a mess. Notes as follows.

* On the SDM4, the side switch should be set at 0 as the code says to open serial port 32, which is Campbell Scientific’s *mumble mumble* weird this means that. Example below.
  + SerialOpen(32, 9600, 16, 0, WS\_BUFFER)

Load the original STTR code onto the CRX

Connect the IMU to the CRX. Data should be coming in

C7 Orange

C8 Yellow

12v Green

G Purple

**April 29th, 2021 – Putting the three sensors together and HYG10 wiring**

Program: 2021.4.29 HS50, MS3, IMU Together

Fixed adding the other scan command for the MS3 by adding the code below, as the MS3 scan requires three seconds, which doesn’t jive with the previous scan (thank you to [here](https://www.campbellsci.com/blog/prevent-skipped-scans) for the code)

NextScan

SlowSequence

Connecting the HYG10 Made code for this using Short Cut, so will test that out next week

12v Brown

C# White (must be C1, 3, 5, or 7)

G Black

Blue Ground (if present)

Gray Ground (if present)

May 6th, 2021 – Reconfiguring Data Tables

TableFile ("CRD:MS3\_Day",64,-1,0,5,Min,0,0)

May 14th, 2021 – Connecting PAR Sensor, finding code, how to connect

Program: 2021.5.14 PAR Sensor

Command was found to be “VoltDiff”. See PC400 CRBasic Editor Help window for specifics.

VoltDiff(PAR\_Den,1,AutorangeC,1,True,0,60,1,0)

VoltDiff(Dest, Reps, Range, DiffChan, RevDiff, SettlingTime, fN1, Mult, Offset)

Connecting the Li-Cor PAR sensor, atmospheric:

These sensors will connect to the differential channels on the CRX, the pair specified in the code in the DiffChan area of the VoltDiff line of code. The code specifies differential channel #1 for this sensor.

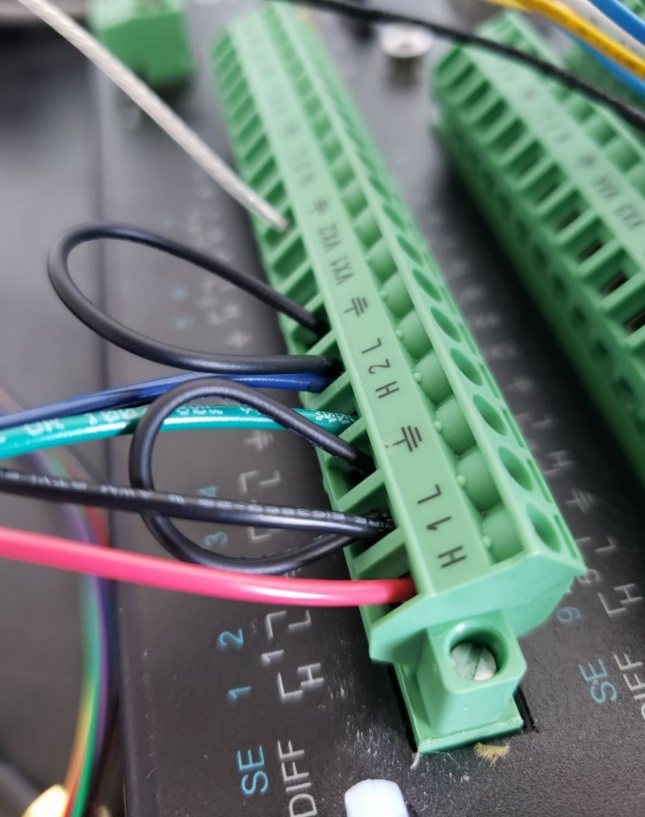
Red Common High (H)

Black Common Low (L)

Connect a jumper wire between the Common Low and ground

May 17, 2021 – Connecting Aqua PAR Sensor

Program: 2021.5.17 Three plus two PAR

Connecting the Li-Cor PAR Underwater Sensor:

Differential channel #2

Green Common High (H)

Blue Common Low (L)

Connect a jumper wire between the Common Low and ground

Same code as before, except everything is labeled PAR2 instead of PAR (VoltDiff, Public, Units, DataTable).

Further questions:

What do the values coming into the CRX mean? Are they in fact micromoles and if so how do we convert them to watts per meter squared? Can the same code connect with the other sensor? What does the sensor BNC to wire connector do, in regard to the impedance in OHMs? Does that mean we need to calculate using the formula in the manual to find the multiplier to put in the code? How does the multiplier work?

IMU Axis Labels

Cables leaving the device is the forward end, GNSS antenna is the after end

Yaw – Z axis, clockwise is +

Pitch – Y axis, cables up is +

Roll – X axis, facing forward, clockwise is +

May 18th, 2021 – TableFile data format, PAR\_aqua2 code, and data file names

2021.5.18 Three by Three, option 8

Specified option 8 in the TableFile command to have the CRX make an ASCII file, instead of a Campbell Scientific proprietary file. Option 8 is TOA5, Header, TimeStamp, Record# (check the CRBasic Editor help area)

TableFile("CRD:PAR\_aqua2\_hr",8,-1,0,60,Min,0,0)

Updated the file names for the sensor respective DataTable to better reflect their names.

Relevant part of the SDM-SIO4A manual, regarding COM addresses for code:

*The SDM-SIO4A has a single address switch. This sets the address of the first serial port of the module (port 1). Other ports have their own SDM address which follow sequentially from the port 1 address. For example, if the address switch is set to 4, port 1 has SDM address 4, port 2 has SDM address 5, port 3 has SDM address 6, and port 4 has SDM address 7.*

Meaning, the HS50 is connected to port one of the SDM4 and accessed in the code via the command SerialOpen(32, 9600, 16, 0, WS\_BUFFER, 0)

Further ports on the SDM4 are opened simply by with the following:

SDM4 Port 1 SerialOpen(32, …)

SDM4 Port 2 SerialOpen(33, …)

SDM4 Port 3 SerialOpen(34, …)

SDM4 Port 4 SerialOpen(35, …)

September 19th, 2021 to Sept 26th, 2021 – EXO code, connections

*Attempted RS-232 to no success.*

Have been attempting to connect the EXO2 via RS-232 to no avail. After talking with CampSci engineers on the phone completely scrapped trying RS-232; moving to SDI12 after learning more than one sensor can be used on an SDI12 port, just need to know the sensors address.

DCP (Data Collection Platform) Signal Output Adapter LED status meaning (pg. 34 manual)

Off No Power

On, no flash No Sonde connected

Flashing 1Hz Sonde connected, everything normal

Flashing 0.1Hz Low power sleep

Sept 27th, 2021 – Talk with CampSci. Talked with Brian Olsen: Case Number: **145266**

1. Connect to datalogger
2. Open the terminal emulator
3. Open terminal
4. Hit enter a few times
5. Type h and hit enter (help command, shows list of commands)
6. Sdi12 talk
7. Enter corresponding port
8. Type command “?! then hit enter
9. If it returns the number one that means it’s address is one
10. 1I! (that’s [1[capital i]! with an uppercase i, not a lowercase L
    1. Sensor should respond with an identifying thing, make mode, revision model, serial, etc. in order to double check you're talking to the right one
11. if you want to change the address the syntax is “1A0!
    1. Explained as [Current address][A][Desired address]!
12. In the code confirm that each sensor is queried at the correct address
    1. SDI12Recorder(SDI12(),C7,"0" (the last one is the address)

9/28 Scotty YSI said no to them having code, but he might get back to us

C7 is Tx, C8 is Rx.

October 4th, 2021 – More EXO fiddling

Set data interval on EXO2 to 3 seconds via KorEXO software

Can connect to the EXO2 via the Terminal Emulator in the PC400 software, following directions above.

?! – confirmed the SDI12 address is zero, talked with the sensor. Can change address.

0M! – The start measurement command will show the countdown till measurements are taken, but it doesn’t look like three seconds…

October 5th, 2021 – Ensure correct SDI-12 sensor address

?! – returns address 0

0I! – returns 013YSIIWQSGEXOSND100

Address and sensor ID are correct.

October 11th, 2021

*2021.10.11 Exo Code v14 SDI12 v4*

The EXO now sends data to the datalogger at 2 minute intervals (yay!). I believe (after stopwatch timings and analyzing data table time stamps) the following is happening: 2sec awake, 1sec sample, 30sec wipe, 30sec awake, 57sec asleep (equals 2min total)

Sonic Anemometer returns

013GillInst 266920019470027

Later in October…

*2021.10.18 Final v1*

Now that the MS3 and EXO have their own SDI-12 addresses, put them on the same COM port – C5. Specify in code the correct COM port and SDI-12 address for each sensor.

Connected the EXO with everything and it is collecting data at 4 minute intervals. After consulting with Brian Olsen, the Camp. Sci. Tech, the issue was identified to be the slow sequence simply cannot run at a 2 minute interval with the main scan busy at 40Hz. Brian: “Remember that the main scan has priority over slow sequences and that slow sequences are executed only when the CPU has ‘spare’ time.”

The way to check how long a scan takes to run is to look in the Status Table at the value of SlowProcTime(1). Brian: The value is the amount of time it takes for the slow sequence to run in microseconds. Thus, a value of 1,000,000 is 1 second and 120,000,000 is 120 seconds, or 2 minutes. If the value is greater than 120 seconds, then the slow sequence simply cannot run at a 2 minute interval.”

The slow sequence takes 186 and change seconds to run. If I set the slow sequence to 3.5min (or just 4min) I should be okay.

Brian: “Change the last parameter of the SDI12Recorder() instruction for the EXO from 1 to 0. This will cause the program to NOT wait until the measurement is complete before proceeding to the next instruction. The data will be collected on the next scan. This will likely mean that the slow sequence can run in less than 2 minutes, but the data in the EXO table will be “late”. In other words, the data that is timestamped in the EXO table as “2021-10-18 12:08:00” are data that really were taken 2 minutes earlier.”

November 8, 2021 – Final EXO touches and Power Draws

*2021.11.9 Final v3*

|  |  |  |
| --- | --- | --- |
| **Sensor** | **Max** | **~Average** |
| EXO | 0.11A | 0.04 |
| R350 | 0.18A | 0.18A |
| IMU | 0.6A | 0.04A |
| MS3 | 0.9A | 0.06A |
|  |  |  |

Following Brian’s advice changed everything back to 2 minute intervals with a 0 in the last position of the SDI12Recorder instruction to NOT wait on sensor timeout.

February 1st-22nd, 2022 – Connecting to Cell210 Module

*no code – see Quick Deployment Guide for Cell 210-Series*

Following Instructions for Call 200-series Quick Deploy Guide

* When connecting through the CS I/O port, power for the module is provided by the data logger (pg. 31 and in quick-start guide)
* Connect the cellular antenna to the Primary Antenna connector. Mount the cellular antenna so there is at least 20 cm between the antenna and the user or any bystander. (pg. 32).

1. Go to [www.konectgds.com](http://www.konectgds.com) and make an account
   1. User: [buoy@csum.edu](mailto:buoy@csum.edu), pass: Buoy4!ocean
2. Ensure sim is installed in the cell module, connected to datalogger and antennas
3. Click Devices and Services, Redeem PakBus Router Code, and enter router code
   1. 5XKP065DMV is our router code, found on the paper accompanying the cell module
4. On the following page it gives the following information
   1. DNS: jaros.konectgds.com
   2. Port: 8650
   3. TCP Password: [blank], I entered “Buoy4!ocean”, same as before
   4. PakBus Address: 1
   5. *When using LoggerLink please use a Neighbor Address of 4070*
5. Configure the datalogger in Device Configuring Utility (Section 5.2 of Quick Deploy Guide)
   1. Connect using Device Config. Utility
   2. On **Datalogger** tab, change the data logger **PakBus Address**and optional **PakBus/TCP Password** to match values entered in the Konect PackBus Router setup.
   3. On the **Network Services** tab in the **PakBus/TCP Client** field, enter the DNS address and Port number noted during the Konect PackBus Router setup.
   4. On the **PPP** tab, set **Config/Port Used** to **CS I/O SDC8** or **RS-232**, depending on how you are connected to the datalogger.
   5. (Optional) On the **PPP** tab, set **User Name** and **Password** if required by cellular carrier (usually outside the U.S.).
   6. Verify the **Modem Dial String** setting is blank.
   7. If connected through RS-232, on the **Comport Settings** tab, set **RS232 BaudRate** to **115200 Fixed**.
   8. Shut down Device Config. Utility and start it again. This will activate the **Cellular** tab needed for the next step (If doesn’t appear close everything, restart computer).
   9. On the **Cellular** tab, enter the **APN** provided by your cellular provider (located on the Cellular Data Service Provisioning Report, from Camp. Sci. in our case). Click **Apply** to save changes

DO NOT set the Config/Ports Used setting to USB. DO NOT DO IT. If you do see below instructions.

HIDDEN WAY TO RESET THE CR1000X (had to call, not in public facing support documents):

1. *Disconnect power supply from ALL power – USB and power supply*
2. *Open Device Configuration Utility, go to “Send OS” tab*
   1. *Click “Start”*
   2. *Select the operation system for the CR1000x, in my case “CR1000X.Std.04.02”*
   3. *DO NOT hit open yet, just select the file*
3. *Hold down the eject button on CR1000x*
4. *Plug the CR1000X into the computer via USB*
5. *Wait a moment for it to initialize/connect*
6. *Click “Open”. It should start sending data to the CR1000x, green bar across the bottom. Release the eject button and let it finish. Once it finished it reboots itself, so give it a minute or two, then try reconnecting.*

Unanswered questions:

* Make up to four trusted IP addresses to check in with the buoy, since it will accept all incoming IP addresses unless you specify trusted sources.
  + Connect via DevConfigUtil, Settings Editor, Network Services, Trusted hosts “edit”

(Optional) By default, the CELL200 series will accept incoming communications from any IP

address. This can be a security risk. You may specify up to four IP addresses, with wild cards,

to limit connections to only those trusted sources. Use an asterisk (\*) as a wild card. For

example, a setting of 166.22.\*.\* would allow connections from devices that have IP

addresses starting with 166.22. Both IPv4 and IPv6 addresses are supported

Next steps:

Set up LoggerNet: this is the next step in the quick deployment guide, but are we using that?

Test of buoy internals outside on the cart

February

*no code*

PG 48 manual

Activating the wiper: The EXO2/3 system is likely equipped with an central wiper to clean the sensors. There are two different

mechanisms to activate the wiper.

The first is to write any number into register #3, this will trigger the EXO sonde to wipe the sensors in both directions. 60 seconds should

be allocated for the wiping to complete, and the data presented to the Modbus holding registers during the wiping sequence will

not be representative of the water quality because of the effects of the wiper passing over the sensors. It may be helpful to program a

routine wipe interval into the SCADA system as well as an operator button to manually trigger the wipe sequence.

The second method is to program the sonde to autonomously sample at an interval that is greater than every two minutes. By default the

sonde will wipe all the sensors before taking a reading. So programming a 1 hour deployment in the KorEXO software the sonde with

automatically wipe the sensors. Note the real time data presented over Modbus during the wiping sequence will not be representative

of the water quality because of the effects of the wiper passing over the sensors. Thi

WRITTEN WHILE ATTEMPTING TO CONNECT THE EXO VIA RS-232

~~It has to be connected through RS232, as all the COM ports are used on the CR1000X~~ is FALSE

MUST figure out how to use the: SerialInRecord or SerialIn

September 20th, 2021 – EXO code, connections

Attempting using this command today:

SerialOpen(32, 9600, 16, 0, 80)

SerialOpen(COM,Baud,Format,TXDelay,BufferSize

Format - Logic 1 high; No parity, one stop bit, 8 data bits; No error checking; PakBus communication can occur concurrently on the same port

SerialInRecord(32,EXO(24),0,80, &hend, iNWSRead, 01)

SerialInRecord(COM,Dest,BeginWord,NBytes,EndWord,NBytesReturned,SerialInRecOption

SplitStr

FilterString

SplitOption

The order of the RS-232 parameter output is controlled by the SDI-12 tab on the deployment menu.

Table

Description automatically generated

SerialIn(EXO(6),32,0,33,0)

SerialIn(desitination,

THURSDAY –

HS50 config (Terrusa and ACL)

EXO2 config (Kate, ACL, Terrusa)

Put in box, clean cables (Sophie)

Rename HS50 to R3-50

Integrate EXO into the datalogger software

Units for things

Transects – 1 flood, 1 ebb, 1 slack, 1 high tide, 1 low tide) and most importantly water level

* Water samples (Dr. Parker)
  + TSS (Total Suspended Solids) and chlorophyll
* Depth measurements (echo sounder)
* Water quality meter (EXO)
* Qualitative observations

Next steps

* ~~Run the sonic as a 232 instead of 422~~
* ~~Data table formats for each sensor for 1 hour individual files~~ 
  + Config output files?
* ~~Time stamps for all data (CRX or IMU) (Terrusa found out how).~~
* ~~How does the SDM4 work?~~
* Put everything on the tower/mast
* 72 hour test on the roof outside (or inside on the bench?)
* ~~Add the HYG10 to the datalogger~~
* Power consumption of each sensor and datalogger (voltage, current (amps)
* Excel sheet
  + Instrument – type (atmo, sub surface) – communication – voltage, current in amps
* Test IMU to find coordinate system for pitch, yaw, and roll
  + Positive, negative, etc for each axis

DataTable (FifteenMin,True,-1 )

DataInterval (0,15,Min,10)

In addition to the tables below, refer to the Public and Dim instructions, which are used to define variables in the program, as well as the data table output instructions and string instructions for information on defining and using a specific data type.