Issues:

Make sure GPS data will translate correctly to eastern hemisphere.

Double check “local time offset”

1

2

3

5

8

4

6

7

Female plug

1

2

3

4

5

6

Female plug

Garmin 18 LVC

Connection of the GPS to a laptop with USB can be done with a PL2303 or a TU58, which I have about three. The GPS should be powered with 5V, and will produce a +/- 5V signal on the TX line. Connect GPS GND & TX to the GND & RX pins on a DB-9 connector, which is then connected to the RS-232 to USB adapter.

A good serial software program is XCTU, which I’ve saved a copy on the thumb drive. Tools->Serial. Default config for the GPS is 4800-8-N-1-N. A rush of $GG messages should stream by.

(One of the older Prolific PL2303 USB to serial driver. Needs an old driver installed, v. 3.3.2.105 which is saved on the thumb drive.)

Basic connection of the Garmin

6 -pin

|  |  |  |
| --- | --- | --- |
| 1 | yellow | pulse |
| 2 | red | Vin |
| 3 | black | ground |
| 4 | white | TX |
| 5 | black | ground |
| 6 | green | RX |

There is Garmin software to configure the GPS (Appendix E) but I was not able to connect with my comm port.

Communicating with the GPS via a serial console:

Enter this command to get sensor config information (4.1.3): **$PGRMCE<CR><LF>**. Can add hex ‘0D0A’ to a send string: **24 50 47 52 4D 43 45 0D 0A**

The initial return value

$PGRMC,A,00220.5,100,6378137.000,298.257232666,0000,0000,0000,A,3,1,2,04,30.0\*71 (A is field <9>)

* The default values are good for now. Check on altitude above sea level.

Additional sensor config info (section 4.1.4)

$PGRMC1,1,1,2,,,,1,A,N,,,,1,2,2\*6B

Output sentences (section 4.1.5).

Want to disable three sentences: GPGSA, GPGSV, and PGRMT leaving only GPGGA and GPRMC which is what the data logger is expecting. ~~Also want to change the baud rate to 38400 for the Campbell GPS() function.~~  Can’t do it.

Changed baud rate on the GPS to 19200 because 38400 is not doable on this GPS.

**$PGRMC,,,,,,,,,,5,,,,30** The last parameter was required to get the command to take hold.

Gps\_only.CR6 works. It takes a bit of time for the clock to sync, but it does. It did not sync at 4800 but that may have also been a wire issue (?)

EXO Sonde

Wire connections to SOA-DCP

|  |  |  |  |
| --- | --- | --- | --- |
| Pin | Luke’s | YSI official | SOA-DCP |
| 1 | black | orange | n/c |
| 2 | white | black | BLK - gnd |
| 3 | red | red | RED - pwr |
| 4 | green | bare | BARE - shield |
| 5 | orange | yellow | YEL - comA |
| 6 | blue | white | WHT - comB |

Connecting the mini USB to a laptop and running KorEXO works just like the blue serial adapter. The sonde can be programmed. No need for power other that EXO batteries and USB power.

Use the DB-9 to screw terminal adapter along with the black Benfei cable to communicate with the sonde via a serial terminal program (XCTU). The SOA-DCP responds to commands from the SOA-DCP quick start document. You need to power the SOA-DCP separately, and get the EXO to respond to commands (e.g. “data”). *It doesn’t just spit data out*. Commands are terminated with 0x0D. Sometimes the command needs to be sent 2 or 3 times. It can be a bit flaky. The SOA-DCP should respond by blinking 1/sec when it’s awake.

|  |  |  |  |
| --- | --- | --- | --- |
| SOA-DCP | color | Test setup | CR-6 |
| VDC-in | red | 12V | 12V |
| GND | black | GND -5 | GND |
| 232-TX | blue | RX - 2 | C4 |
| 232-RX | white | TX - 3 | C3 |

A Flame test deployment template was created in KorEXO. The type and order of variables was taken from Super FLAMe code. **Before testing with the CR-6, modify SOA-DCP by changing baud rate to 38400 and set string delimiter (setdelim) to comma (2), and set PWRUPTORUN = 1.**

setcomm 7 1, setdelim 2, pwruptorun 1

**Suna**

Table

Description automatically generated

The included CD didn’t make much sense to me, but I can connect to the sensor via serial (USB or RS-232). Params are 57600/8/N/1/N.

Section 6.2 of the Suna Users Manual goes into serial communication including the commands.

Termite is a good program to use. Need “set” and “get” prefixes with the command to get them to work (not mentioned in the manual)

|  |  |  |
| --- | --- | --- |
| Pin | Luke’s | Suna |
| 1 | black | Vin |
| 2 | white | Ground |
| 3 | red | USB 5V |
| 4 | green | SDI-12 |
| 5 | orange | RS-232 TX |
| 6 | blue | RS-232 RX |
| 7 | white/black | Analog Vout |
| 8 | red/black | Analog Current |

**LGR & pressure**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Pin | Luke’s | RS-232 | Converter  In | Converter  Out | CR-6 | ILLINOIS |
| 1 | black | Gnd (5) | 5 |  |  | Gnd (black) |
| 2 | white | TX (3) | 3 | IN | U9 | TX |
| 3 | red | NC |  |  |  |  |
| 4 | green | NC |  |  |  | Vout (green) |
| 5 | orange | NC |  |  |  | Agnd (orange) |
| 6 | blue | RX (2) | 2 | OUT | U10 | RX blue |
| 7 | w/b | NC |  |  |  | Dgnd |
| 8 | r/b | NC |  |  |  | Supply (12V) |

Illinois – blue are PTB110 pins

8

7

6

2

3

4

5

1

Female plug

Com port set to 9600-8-N-1.

5V VCC for the serial converter unneeded. It outputs only 0-4V.

RX – 2

**Garmin Depth Sounder**

|  |  |  |
| --- | --- | --- |
| Cable | CR-6 (Reno) | CR-6 (Green) |
| red | 12V | 12V |
| black | Ground | Ground |
| blue | U4 | NC |
| brown | NC | U4 |

These messages are activated in the GUI and used to capture variables: SDDBT, GPGLL, GPGGA, SDMTW. Serial data transfer settings are 4800,8,N,1. Test program is *echo\_sounder\_only.CRG*

**CDOM/PeakT**

FLAME connections to CDOM/PeakT

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Pin | Turner | Function | CR6 - CDOM | CR6 - Peak | IL - CDOM | IL - PeakT |
| 1 | Red | Power | 12V | 12V |  |  |
| 2 | Black | D-Ground | Gnd | Gnd |  |  |
| 3 | Org/Wht | Signal Out | U5 | U7 |  |  |
| 4 | Green | A-Ground | U6 | U8 |  |  |
| 5 | Blue | X10 | U6 | U8 |  |  |
| 6 | Yellow | X100 | NC | NC |  |  |

CR-6 Issues

Comm memory: Inadequate memory in the SerialOpen command causes regular data freezes. Increased from 800 (SuperFlame) to 1200. Check status=CommsMemFree for available communication memory.  *SerialOpen(ComU9,9600,16,0,1200,1) ' U9/U10 for GGA as 5V TTL*

Sections 8.3 (SRAM) describes how to increase comms memory:

Connect -> Datalogger -> Settings Editor -> Advanced -> Communication Allocation

FLAMe-IL issues

SDM:

Multiplexer: C1=enable; C2=clock; C3=data; Data logger: C5=data; C6=clock; C7=enable

SC105 -- (CampbellSci CS I/O to RS-232 Interface document)

Configuration setup: The ‘datalogger’ side is connected to the CS I/O port for power. The ‘modem’ side is connected to the PC via a TU-S9 converter.

Open a serial program. Pressing the button on the side for 1sec brings up the menu.

Configuration selections:

1. CS I/O Port = Modem Enable
2. RS-232 Port = 38400, 8, N, 1 (for EXO only)
3. RS-232 Port = 9600, 8, N, 1 (for LGR only)

Note: null modem required: cross the TX/RX lines

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| SOA-DCP | DB-9 |  |  |  |  |
| TX | 3 |  |  |  |  |
| RX | 2 |  |  |  |  |