**LARA mooring (4/28/2018)**

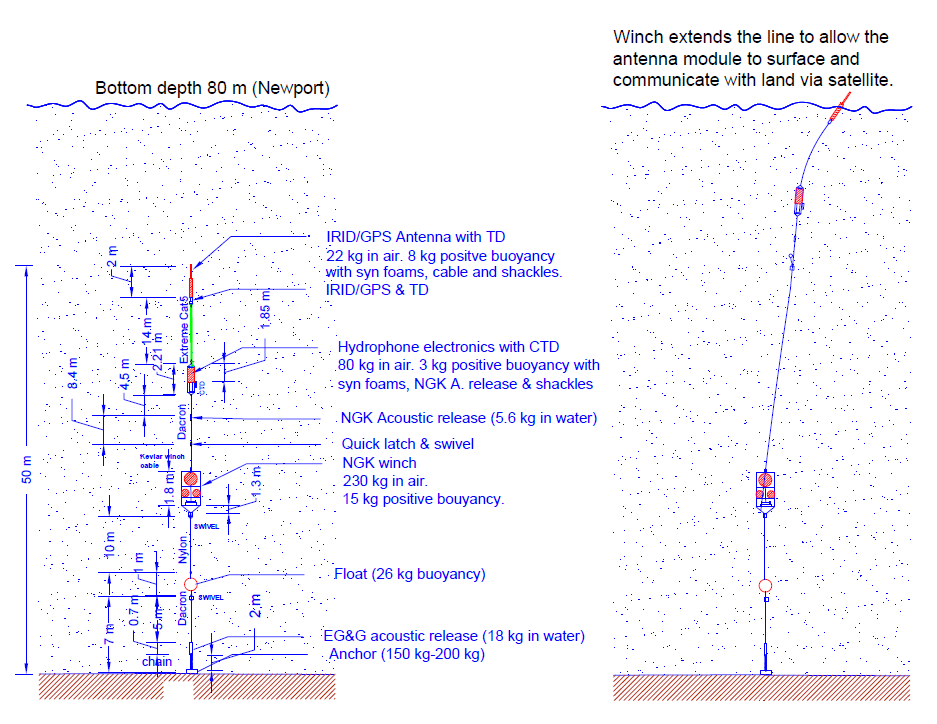


Figure 1. Newport mooring

LARA is an underwater winch system with a passive acoustic data logger/detector. It is capable of monitoring the acoustic activities where the seasonal sea-ice covers the surface. Once a day the winch raises the antenna to the surface for data communication. It consists of three subsystems: a) an underwater winch from NGK, b) buoy electronics with passive acoustics system and CTD, and c) antenna with TD sensors. The buoy electronics and winch system are physically connected by a Kevlar line (250 kg max load) but not connected electronically. Between the buoy and antenna electronics are connected by a 15-m long multi-wire Extreme CAT cable. The winch has its own controller with 1-year life battery, which can be controlled remotely by communicating over the acoustic modem. The buoy electronics includes the high performance hydrophone system (WISPR), CTD, and acoustic modem. The top two subsystems (antenna and buoy) communicate over RS232 serial lines. Current prototype system uses an alkaline battery package with 16.5 V output and 135 AHr (@15 AHr per column (@10 °C).

The max length of the Kevlar cable between the winch and buoy is 150 m. 95 % of the day the entire system stays below the surface (>60 m) to avoid the rough weather near the surface. 5 % of the day the top two systems reaches near the surface for data transmission. The depths of the system are monitored by at two sensors: one at the buoy electronics (SBE16Plus) and the other at the antenna electronics (SBE39Plus). Having two depth sensors allows estimating the current speed by the tilt angle. If the tilt is extreme, e.g., 30 º, it can cancel the ascent. A few test deployment would provide data what angle to stop ascent.

If one of the two sensors detects temperature below -1.79 ºC, the surface may be covered by sea ice, and the ascent is aborted and retracts back to the top two packages to the normal depth (<60 m) to continue the acoustic monitoring. This function, however, not possible to test off Newport and has to be tested in the future deployment near arctic.

For Newport tests, since they are short tests, two WISPRs are sufficient. That leaves two unused WISPR COM ports. 6-pin underwater connectors are connected by six separate wires between the antenna and buoy electronics, including ground, 12-V PWR, two sets of RX and TX lines. We will use one WISPR COM port, namely WISPR4 to communicate directly with the A3LA-RG Iridium/GPS modem in the antenna unit and the other set of RX/TX to communicate with CF2 of antenna unit.

The buoy system comprised of: HTI92WB hydrophone. EOS pre-amp, analog multiplexer, multiple WISPRs (two for Newport test), MPC with CF2 microprocessor, battery package, SBE16Plus CTD with Fluorometer from SeaBird, and acoustic modem (AMODEM) from NGK. There is no electrical connection between the buoy and the winch. To communicate with the NGK winch which could be 14 to 150 m in distance, buoy uses the acoustic modem to communicate.

The antenna system consists of: Iridium/GPS antenna, uMPC with CF2 to control the antenna switch, and the SBE39Plus TD. The antenna system receives the power from the buoy (voltage dropped to 12 V by DC-DC) through Extreme CAT cable. The antenna system’s CF2 has to recognize when to turn on/off the IRID/GPS modem, switch the antenna switch, and sends TD data back to the buoy electronics. The buoy and antenna systems communicate via two separate serial lines: one to directly talk to the serial port of IRID/GPS modem and the other to talk to the CF2 on uMPC.

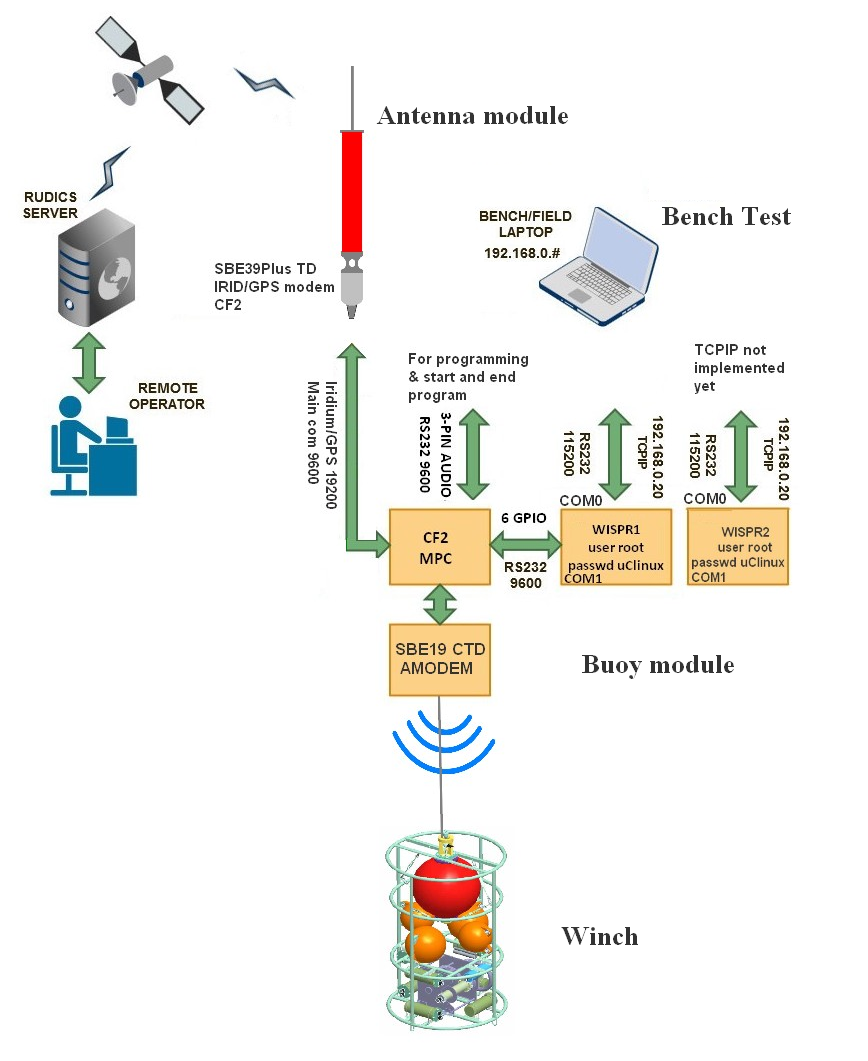
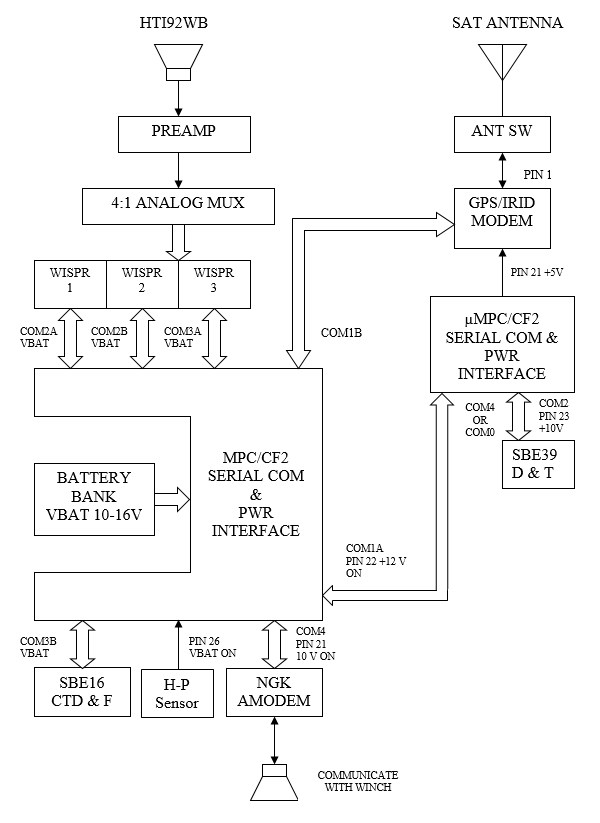


Figure 2. LARA and submodules communication

Figure 3. Functional diagram of LARA system

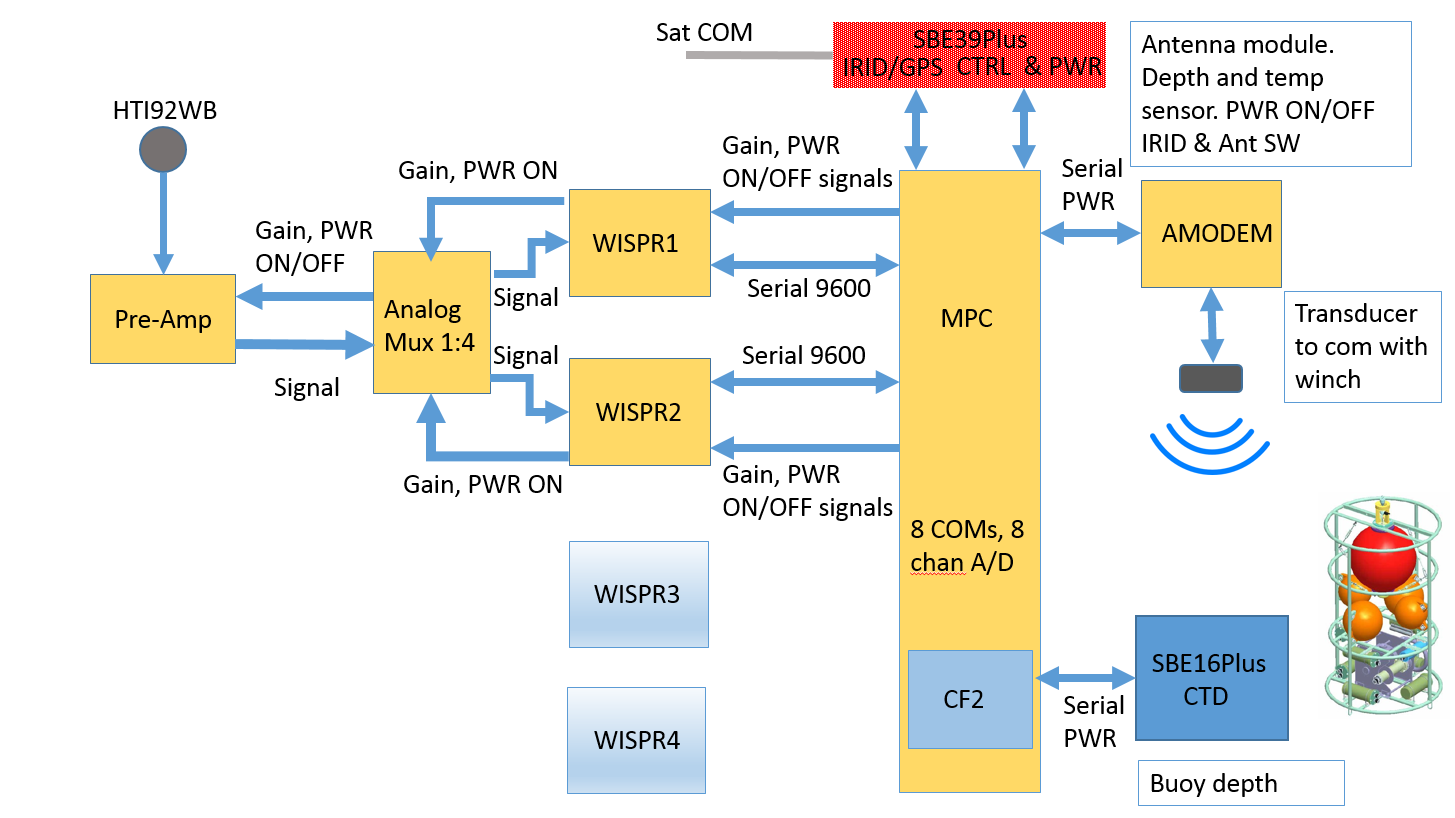


**Buoy System**

**Antenna**

**System**

1. **Buoy module**



**Figure 4. LARA Buoy module block diagram (for Newport test)**

**WISPR board (Rev 1.0)**

The WISPR system is a mixed signal embedded DSP designed for space limited and low power applications from EOS. The motherboard is CM BF527E (64 MB) from Bluetecnix, which is an Analog Devices BF537E Blackfin core module, one ADC signal channel, Ethernet, two RS-232 serial channels, and several external data bus interfaces. The WISPR system also provides an integrated CF Card for mass data storage and an IDE/PATA bus interface for external drives. The system supports software development in the uCLinux operating system (<http://blackfin.uclinux.org>) and VisualDSP++ ([www.analog.com](http://www.analog.com)). The system uses a customized uCLinux kernel that will boot from flash memory on power-up into a fully functional embedded OS with device drivers for the specific hardware and optimized DSP functions.

Up to four WISPR boards can be mounted on Multiple Port Controller (MPC). However for the Newport experiment, our plan is to install 2 since the logging duration is short. The following figure is WISPR board and the table below describe the function of connectors and where they are connected to.

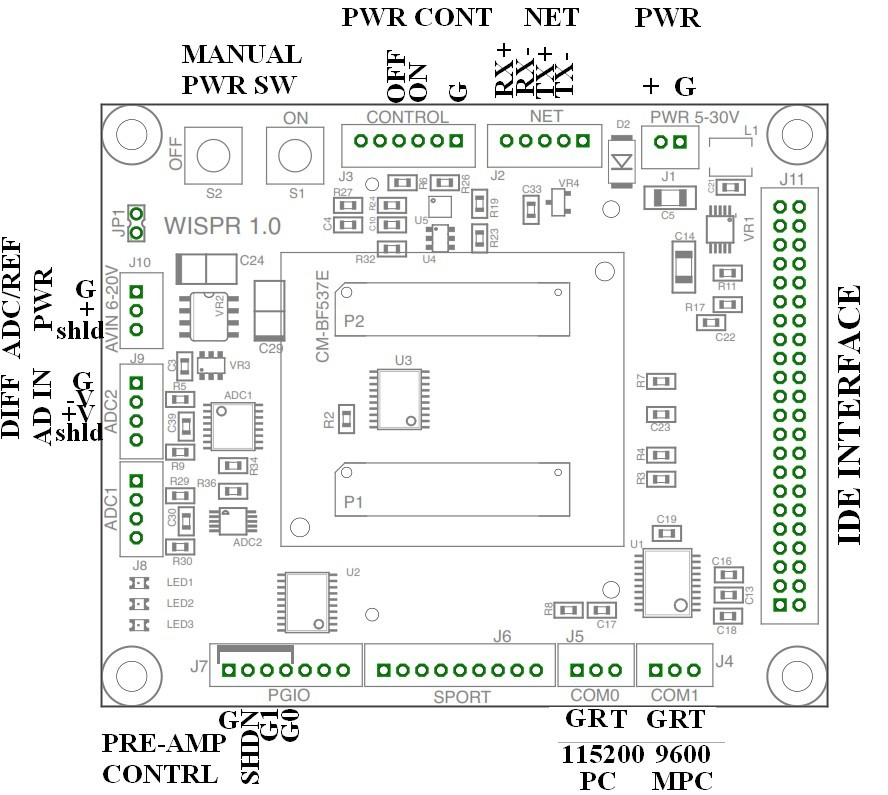


Figure 5. WISPR Rev 1.0

|  |  |
| --- | --- |
| WISRP Connectors | Where to connected |
| COM0 (J5) | Underwater connector “WISPR/CF2 Serial” at the end cap for bench test |
| COM1 (J4) | MPC PAM#C (#=1-4) one of J9C-J12C. Designated for specific WISPR |
| PWR (J1) | MPC PAMPWR# (#=1-4). J9P-J12P Designated for specific WISPR |
| NET (J2) | Not used. Bench test only |
| CONTROL (J3) | MPC PAM#PWR (#=1,2,3 or 4). One of J9P-J12P. 1 ms pulse turns on/off |
| S1 (BLK) | Manual power on (bench test) push button |
| S2 (RED) | Manual power down (bench test) push button |
| AVIN (J10) | MPC PAM#PWR #=1-4, designated for specific WSIPR. |
| ADC2 (J9) | Analog multiplexer output, J5-J8, designated for specific WISPR |
| PGIO (J7) | Analog multiplexer J1-J4, designated for specific WISPR |

Table 1. WISPR connectors

WISPR requires two power inputs: one to power the Bluefin DSP at J1 and the other to power the ADC at J10 separately. Note that ADC input range is 0-5V at J9. It controls the pre-amp power and gain (G0 and G1) at J7. The control signals are sent through analog multiplexer. The WISPR is powered on/off by an external short pulse to J3 pins from CF2/MPC board. It can also be powered on manually by pressing S1 (black) or down by S2 (red). You must be careful not to turn off the power while the logging program is running. Since the files system is still mounted, there is a chance of damaging the file system.

It has two COMs: COM0 (115200 bps) for a console terminal and COM1 (9600 bps) to interface with CF2 through MPC’s PAM serial com port. COM0 of WISPR1 and 2 are connected to one of the MCBH6 connectors at the end cap so that user can monitor and take control WISPR on the bench. Although TCPIP port is there, at present no signal is not connected to the end cap.

**WISPR program**

On CF2 card there should be four files including, console prompt, wispr, start and click.wav. Several versions of WIPSR logging programs have been developed including wispr\_bw (logging and detection program for beaked whale) and wispr\_kw (program for killer whale monitoring). Both wispr programs are a command line program. It logs the acoustic data and runs a real-time detection algorithm for beaked whale (or other species) simultaneously. You can add several options in the command line in the ‘start’ file of the CF card. When the WISPR board boots up, it automatically executes ‘start’ file on the CF card and the last line is ususally ‘wispr\_bw’ or other program.

## The followings are descriptions of the wispr command line parameters.

Options: DESCRIPTION DEFAULT

-M {mode} Processing mode number [ 1 ]\*

-T {secs} Size of ADC data buffers in seconds [up to 8 seconds]

-F {level} Sets flac compression level [ 2, 3]

-s {bitshift} Sets data bitshift [ 8 ]

-o {organization} Organization in FLAC metadata string [ none ]

-b {number} Number of data buffers per file [10 ]

-n {nclick} Min number of click for detection [ 10 ]

-q {nbufs} data snipet option (not implemented yet) [ ? ]

-i {number} Number buffers to skip between file [ 10 ]

-p {prefix} Data file name prefix [ wispr\_ ]

-l {filename} Log file name [no log file ]

-v {level} Verbose level (0=none) [ 0 ]

-L Enable LEDs [ disabled ]

-r Request GPS time, lat/lon and gain at start [ disabled ]

wait GPS signal for 17 sec, for free disk space 57 sec

-W Simulated test detection mode [ disabled ]

-f sampling frequency 62500, 93750, 125000 [125000]

If wispr\_kw (Killer whale) 93750 is default

-g gain 0, 1, 2, 3 (additonal gain with 6dB incr) [0]

-h print help

-C number of files to record

x cpu usage time

One single program can be operated in the following 5 different modes.

\*Modes

- Mode 1: Record continuously, no detection functionality

- Mode 2: Record continuously with detection functionality

- Mode 3: Record intermittently.

Skip specified number of buffers between files.

- Mode 4: Run beak whale detection function only.

Process incoming data continuously with the detection function

but only write a file when a detection appeared.

- Mode 5: Run beak whale detection function and record data intermittently.

Same idea as mode 4 + record data intermittently.

Used to get an idea how many encounters were missed or to monitor noise levels regularly).

\*\*-r Send interrupt to ext processor to receive GPS time, lat/lon and gain during the first 17-second start up, and wait for 5 sec. Default is to receive the GPS time, lat/lon and gain without interrupt. Send % of free space of CF card 40 sec later (57 sec after start up).

\*\*\*Simulated test detection mode (bench test only)

Test the detection performance by overwriting AD buffer content with a short wave file of a specfic species clicks/calls before processing the buffer. File name has to be **bw\_test.wav**. Detection results are stored in **detections.dtx**.

**Note**

When the CF2 board starts uCLinux on WISPR by sending a pulse to pin 3of J3, boot-up process takes ~15 seconds. The last program it executes is a command line, which may look like

/bin/wispr –l wispr.log -v2 –T8 –b15 -M2 -F2 –g0 –r

It logs the activity to wispr.log with a verbose level 2, adc buffer length 8 sec, writes one file at every 15 blocks (file length is 120 sec long) with FLAC level 2, logs data while running detector, sets pre-amp gain to “0” and reads time once if CF2 provides the GPS time at the beginning of the program. Before the start of logging, it sends a free % storage space available of CF card. To calculate the free space of the Lexar 512 GB card, it takes ~55 sec. WISPR then sends the free space info in % to CF2 board.

Clock speed of the WISPR is slowed down to 250 MHz to save battery power (default is 500 MHz). By slowing down to CPU speed, it saves the power by 25% (~75 mA at 250 MHz). This is set up in the ‘start’ script, and here is how it is done.

echo userspace > /sys/devices/system/cpu/cpu0/cpufreq/scaling\_governor

echo 250000 > /sys/devices/system/cpu/cpu0/cpufreq/scaling\_setspeed

WISPR has a 64-MB of RAM. To use its entire 64 MB and avoid memory leak, you need to reserve the memory for the kernel and the user space. You set this up during the Uboot before the 5-sec count expires and stop loading the kernel by hitting a key and type the following in one line

>setenv bootargs root=dev/mtdblock0 rw clkin-hz=25000000 earlyprintk=serial.uarts0,115200 mem=48M max\_mem=64M$# console=ttyBF0,115200

then

>save

>boot

You do this only once. It does not need to be repeated again.

**Files in CF card**

Start – script file

console\_prompt

gaintest - program to test the pre-amp gain

click.wav

wispr\_bw –program to log and detect beaked whale

spectrogram8kHz – program to report the 8-Kz noise level and wind speed

Example

gaintest –g# Set the preamp gain # and hold the gain output for 15 sec. # is the gain value from 0 to 3.

spectrogram –T10 –C12 –v1 –n512 –o256 –g0 -t20 – l noise.log

-Estimate the noise spectrum from a 120 sec time series which consists of 12 10-sec long buffers. Calculate 512 pt FFT with 256 overlaps. Gain set is 0. After the calculation, it waits for 20 sec for CF2 to respond. If WISPR receives WS? From CF2 within this wait period, it returns

NLW\*, %s, %4d,%04d, tmstr, AveNL8kHz, WindSpeed

Time format is "%m/%d/%g,%H:%M:%S"

## Example of start file of CF card on WISPR

/bin/ifconfig eth0 192.168.0.20

#dhcpcd &

cp /mnt/console\_prompt /bin

chmod 777 /bin/console\_prompt

count=10

quit='q'

echo " "

echo "--- Embedded Ocean Systems WISPR V1.0 ---"

echo "Starting recording and detection."

echo "Enter 'q' to stop."

while [ "$count" -gt 0 ]

do

val=`console\_prompt -t 100`

if [ "$val" = "$quit" ]; then

exit 1

fi

count=`expr "$count" - 1`

done

# Uncomment these to cut cpu freq in half

echo userspace > /sys/devices/system/cpu/cpu0/cpufreq/scaling\_governor

echo 250000 > /sys/devices/system/cpu/cpu0/cpufreq/scaling\_setspeed

sleep 1

# clear cache to free memory

sync; echo 3 > /proc/sys/vm/drop\_caches

sleep 1

# Start detection/recording

#echo "WISPR recording and detection started."

cp /mnt/wispr /bin

cp /mnt/spectrogram8kHz /bin/spectrogram

chmod 777 /bin/wispr

chmod 777 /bin/spectrogram

**/bin/wispr –l wispr.log -v2 –T8 –b15 -M2 -F2 –g0 –r**

**mount /dev/sda1 /mnt**

**/bin/spectrogram –T10 –C12 –v1 –n512 –o256 –g0 -t20 – l noise.log**

The third line from the bottom is the logging program. Once the logging program begins, it continues logging until either the data storage spaces runs out or it receives $EXI\* from CF2 to COM1 to stop logging.

After exiting out the program, it remounts the file system, and executes another program. “spectrogram.” It collects time series for 120 sec (T10 is buffer length of 10 sec and for 12 files and runs spectrogram routine with FFT size of 6512 with 256 overlap and writes the noise estimate at 8 kHz with 500 Hz bandwidth. Approximately 120-130 sec later, it prints out time, noise level in dB and wind speed to COM1. Print format is NLW\*,%s,%04d,%04d", tmstr, AveNL8kHz, WindSpeed.

Note: To know what CPU speed it is running, type

cat /proc/cpuinfo

## Talking to the WISPR board

There are two MCBH6 connectors at the end cap. By connecting to one of these connectors, you can talk to either the 1-st or 2nd WISPR directly via 3-pin RS232 serial at their COM0 port with 115k baud rate. COM0 is a regular console terminal. You can also talk to CF2 on the second serial port at the same time. When the power is turned on, WISPR displays uClinux boot up message, executes a script file ‘start’ under /mnt.

For a bench test, a CF2 program, ‘testpamcom’ allows you to specify which WISPR PAM to boot. On the SISPR’s COM0 side, boot-up message starts appearing. The boot-up is a two-stage process. After the first 10 lines or so, bfin\_mac shows up and halts for 5 sec. During this 5 sec rest, you can type any key to stop “autoboot.” By doing so, you interrupt the autoboot process and it is only loaded with a minimum kernel. At this point, a full Linux system is not loaded yet.

During the second boot-up stage, it loads a full uClinux and penguin figure and Bluetecnix sign show up. It mounts the CF card file system and executes ‘start’ file on the card. The start file configures the serial terminal console, sets up the TCPIP (10MB/s), and make the system clock speed to 250 MHz. Near the last line it launches the data logging program ‘wispr’. When ‘wispr’ starts, it wait for 5 sec to allow a user to stop the program if needed. If you type ‘q,’ it exists out of the program to uCLinux. If you successfully exits out, you can execute the Linux command. If you let expires the time out, it starts data logging program.

When bench-testing, and if you wish to set date and time of the WISPR’s RTC manually after uClinux is fully loaded and running, you can set the software clock first by

date –s 2017.07.10-22:20:10

To synchronize the uClinux time to the hardware clock, type

hwclock –w

Typing ‘ps’ shows process ID numbers and can kill the process by Unix kill command. You can also talk to the WISPR by Ethernet. To talk to WISPR by telnet, open Cygwin or MSDOS window to telnet WISPR by

telnet 192.168.0.20

No login user/password is required. But in case it asks you to log in, user name is root and password is uClinux. You can do file transfer using WinSCP of Windows.

After making sure that no logging/detection program is running, change the directory to /mnt to vi edit the ‘start’ file or remove the data files on CF card from the previous field mission. You can find the ‘start’ script file in /mnt (CF card). Examine the last command line very carefully to see if it does the job for you. After exiting out ‘vi,’ move back to the root directory and unmount the file system before powering down.

**umount /mnt**

If the logging program is still running and disperate, you can terminate the program by the Linux kill command. However, a more graceful way to end the logging program is to send $EXI\* to COM1 (9600 bps) of WISPR. You can do this only when the buoy electronics is open. WISPR’s COM1 port is connected to CF2 and not available at the MBCH6 connectors at the cap.

After successfully exiting out the logging program, wait for 10 seconds and push the S2 (red) button on the WISPR to power off. If you are running CF2 program ‘testwispr,’ typing ‘Q’ would kill the power as well. Pushing the black button (S1) would turns on the WISPR is equivalent to send a pulse and ‘start’ file on /mnt is executed. Pressing S2 (red) button is equivalent to sending a pulse to pin 4 of J3. However powering down the WISPR by hitting the red button (S2) without ending the logging program properly is not recommended. It is a Linux system after all and which does not like to be powered off while its file system is still mounted. You must unmount the file system before powering off.

There are other commands characters that CF2 program sends to WISPR’s COM1. These are described in the next section.

In case you need to re-mount the CF card file system, do the followings from the COM0 serial terminal

**/bin/mount /dev/sda1 /mnt**

If you need to fdisk a new CF card in preparation for deployment, do the following on WISPR from COM0 (RS232 115000bps) window or remotely log in through TCPIP Ethernet.

**/bin/fdisk /dev/sda**

**p** to print partitions

If it is 512 GB is formatted properly, it should look like this.

Device Boot Start End Blocks if System

/dev/sda1 \* 1 622268 500167678\* 83 Linux

A new CF card partition may be different from these, and if it is different, delete the existing partition by ‘**d**’

Partition number (1-4) : 1

Add a new partition by ‘**n**’

e extended

p primary (1-4) :

Type **‘p’** for primary

Partition (1-4): 1

First cylinder (1-62268, default 1) carriage return for default

Last cylinder : carriage return for default

**p** to print the partition again

**w to write the partition**

After fdisk you need to reformat the disk in FAT32. If the file system is still mounted, unmount it first, then

**/bin/mkfs.vfat –F32 /dev/sda1**

To power off the WISPR, be sure to unmount /mnt first (wispr program should not be running when umounting).

To download data files from WISPR to your PC, use cygwin window and type (the last ‘.’ means the current PC directory)

**scp root@192.168.0.20:/mnt/filename . or**

**scp filename root@192.168.0.20:/mnt**

You can also use WinSCP if it is installed on your PC. For file transfer it may ask for user name and password. User name is root and password is “uClinux”. WisSCP may be more powerful than cygwin. Keep in mind that WISPR’s TCPIP speed is limited 10 Mbps which is too slow to download the entire directory of a CF card. It also uses battery power. So to transfer the entire data files to another data disk, it is recommended to retrieve the CF card from the WISPR slot and copy directly on your PC directory.

As described earlier, if you have an access to the WISPR’s COM1, you can start, stop and end the program and a few other functions by talking to the WISPR from CF2 or you PC COM port. The followings are the character strings that WISPR expects to see or sends from COM1. Cr is a carriage return.

$GPS ,%ld, %8.3f,%7.3f\* Cr GPS time, long and lat CF2->WISPR

$DX?,%ld,%ld\* Cr Inq detections CF2->WISPR

$DXN ,%d\* Cr Num of detections WISPR->CF2

$ACK\* Cr Send ACK for each line WISPR->CF2

$NGN,%d\* Cr New gain (0-3) CF2->WISPR

$EXI\* Cr End logging CF2->WISPR

$DET,%d\* Cr Detection parameter WISPR->CF2

$DFP\* Cr Inq disk space CF2->WISPR

$DFP,%5.2f\* Cr Reply disk space avail % WISPR->CF2

Once the system housing is closed, you lose a full access to the WISPR. Only way for the WISPR to communicate with outside world is via its 115kbps COM0 serial port or CF2’s serial. $EXI\* is the character sting to WISPR COM1 to end the program.

**MPC Controller board**

1. **CF2 and MPC**

A CF2 microprocessor is mounted on the MPC (Multi-Port Control) board directly on 50-pin and 20-pin headers. CF2’s console port (COM0) is the 3-pin audio plug on the MPC. This is the port reserved for communication with PC for programming or to manual control. Between WISPR and MPC, WISPR uses its COM1 (9600 bps) to communicate with the CF2 through MPC’s COM2 and COM3 during logging. CF2, when it is idle (hibernate mode), draws less than 1 mA (@15V). In a normal PICODOS mode and when no program is running, it draws ~5 mA. Salutation and prompt (C:>) should appear on console when using the terminal emulation program, such as Motocross.

It has four battery power ports (J1 through J4). Each port can sustain continuous current up to 1 A. NGK modem and Iridium (A3LA-RG from NAL) modem has a peak power up to 2 A. So you must use at least two power ports for a safe operation. 3 or 4 power ports are more desirable.

ADC channels

|  |  |
| --- | --- |
| Chan | Description |
| 0 | current |
| 1 | Battery volt |
| 2 | pressure |
| 3 | humidity |

MPC’s ADC input has 8 channels. The first two (ch 0 and 1) are already in use. Ch0 is the MPC’s current and ch2 is battery voltage. To monitor the current usage, P3 on MPC must be shorted. For more details on the CF2 and digital IO pins, see the end of this manual.

**b) H-P sensor board**

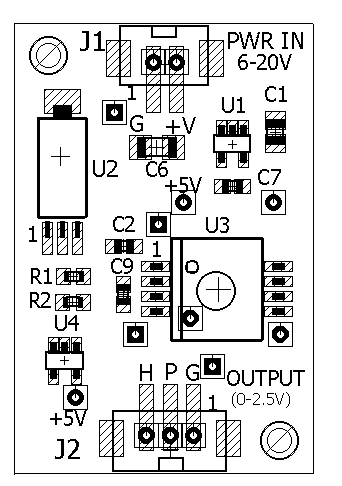
H-P sensor is mounted behind the top end cap by Velcro. It produces analog humidity and pressure outputs within 0-2.5 V range. Pressure voltage is connected to Ch2 and humidity output is connected to Ch3 of ADC input of MPC. Use CF2’s HPSens program in C: display the pressure and humidity after the housing is closed for leak test. Pull the air by a vacuum pump through the PRV port of the end cap to 0.9 atm and monitor the H-P sensor outputs to check for any air leak before the deployment. Run hpsens in C: aaHPSens prints out the pressure in mbar and humidity by %. On the MPC, H-P sensor power is connected to DiFAR power J14. By setting pin 26 of CF2, power of H-P sensor is turned on. H-P sensor board consumes 6 mA at 14 V. So you must make sure to clear the pin 26 to turn the power off by PC 26 after exiting the program by typing “.”

|  |  |
| --- | --- |
|  | Spec |
| PWR Vin | J1 pin 2 6-20 V |
| Current | 6 mA @12V |
| Humidity | J2 pin 2 0-2.5 V |
| Pressure | J2 pin 3 0-2.5V |

HP Sensor spec

|  |  |
| --- | --- |
| CF2 Pin 26 | HP-Sensor board |
| 0 | OFF |
| 1 | ON |

HP sensor logic

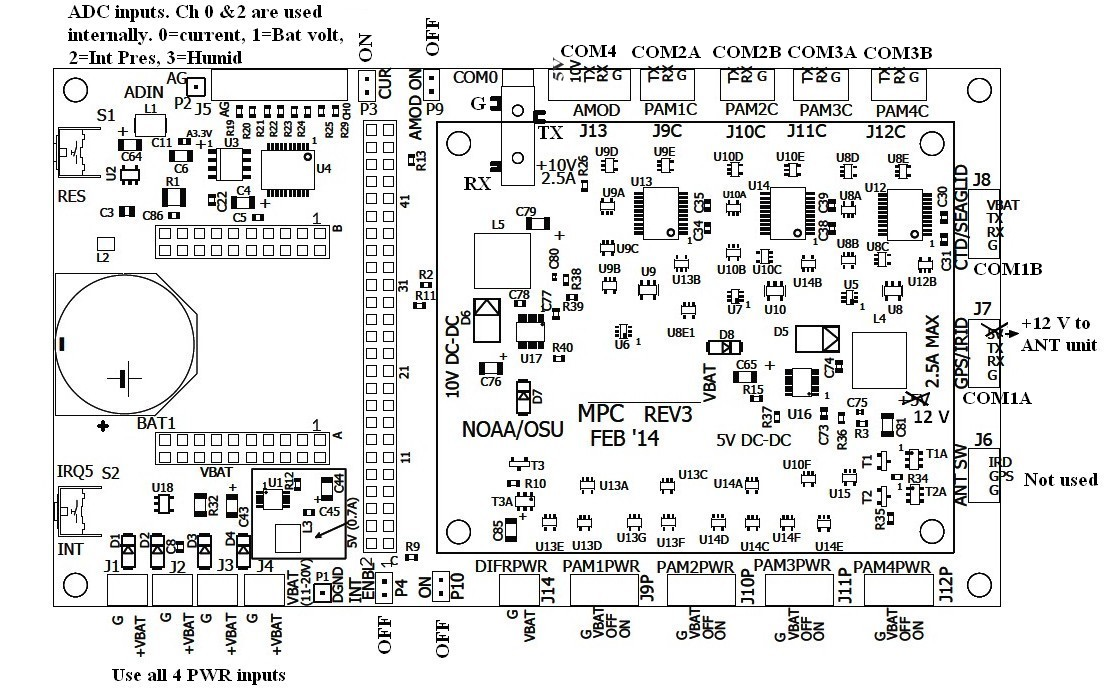


H-P sensor board

Figure 6. HP sensor

**c) MPC board**

With MPC, CF2 can communicate serially up to four WISPR PAM boards through COM2A, 2B, COM3A, and 3B). For the Newport test only, PAM4C is used to communicate directly with A3LA unit on the antenna module. PAM powers are controlled by GPIO lines (PAM1PWR-PAM4PWR). In the CF2 program, you must set four TPUs, 3, 4, 8, and 9 (see Table 1-6 and 1-7) to turn one of the WISPRs’ power and RS232. All the WISPRs’ powers stay connected to the battery while drawing small current, approximately 2 mA each. It remains a low power mode until the appropriate TPU pins are set and receives a short pulse pin 3 of J3. MPC uses TPU 15 to send a short pulse to start uClinux of a specific WISPR board. To shut down uClinux, MPC also sends a short pulse on MODCK (pin 42) to end uClinux on a specific WISPR. The power to WISPRs is maintained at low power mode after the shut-down pulse is sent. MPC communicates with a single WISPR at a time over one of the four COM ports (COM2A, COM2B, COM3A and COM3B). See table 1-3.



SBE16. PWR from J8

To antenna COM.

12V to antenna VBAT

To COM3 of A3LA in antenna

Figure 7. MPC (rev3 with modification at pin 23. R15 is removed.)

Buoy and antenna modules communicate over the 15-m Extreme CAT cable. MPC uses two COM ports: PAM4 (COM3B, J12C) for A3LA and IRID/GPS port (COM1A J7) as a main. For the Newport test, we sacrificed the last PAM port to communicate with the Iridium/GPS modem (A3LA-RG) in the antenna unit directly. We also use the IRID/GPS (COM1A J7) of MPC as a main to talk to the CF2 on the uMPC in the antenna module to power and control the SBE39plus sensor and to toggle the antenna switch by a pulse less than 100 ms pulse.

Note that MPC’s COM1A J7 power output is increased from 5 V to 12 V for the antenna module. Antenna switch port (J6) of the MPC of the buoy is therefore no longer used. As are result one GPIO pin /DS (pin 1) is still available for another application.

**d. NGK Amodem (NGK)**

MPC communicates with NGK acoustic modem (Amodem) serially J13 (COM4). Amodem’s transmitter and receiver board uses 10-14 V and 7-10 V respectively. For a simplicity, MPC sends +10V (2A) power from AMOD (J13) port to the boards. It also requires a +5V line to enable its port. +5V comes from CF2’s 5V DC-DC power supply which is always on. Instead of toggling +5 V on/off, MPC uses the +10-V power to turn on/off. Check the wiring diagram below in order to properly connect the Amodem. To power up, set CF2’s pin 21 (PCS0) high. +10-V power comes up. Amodem uses COM4. Amodem station ID is 02.

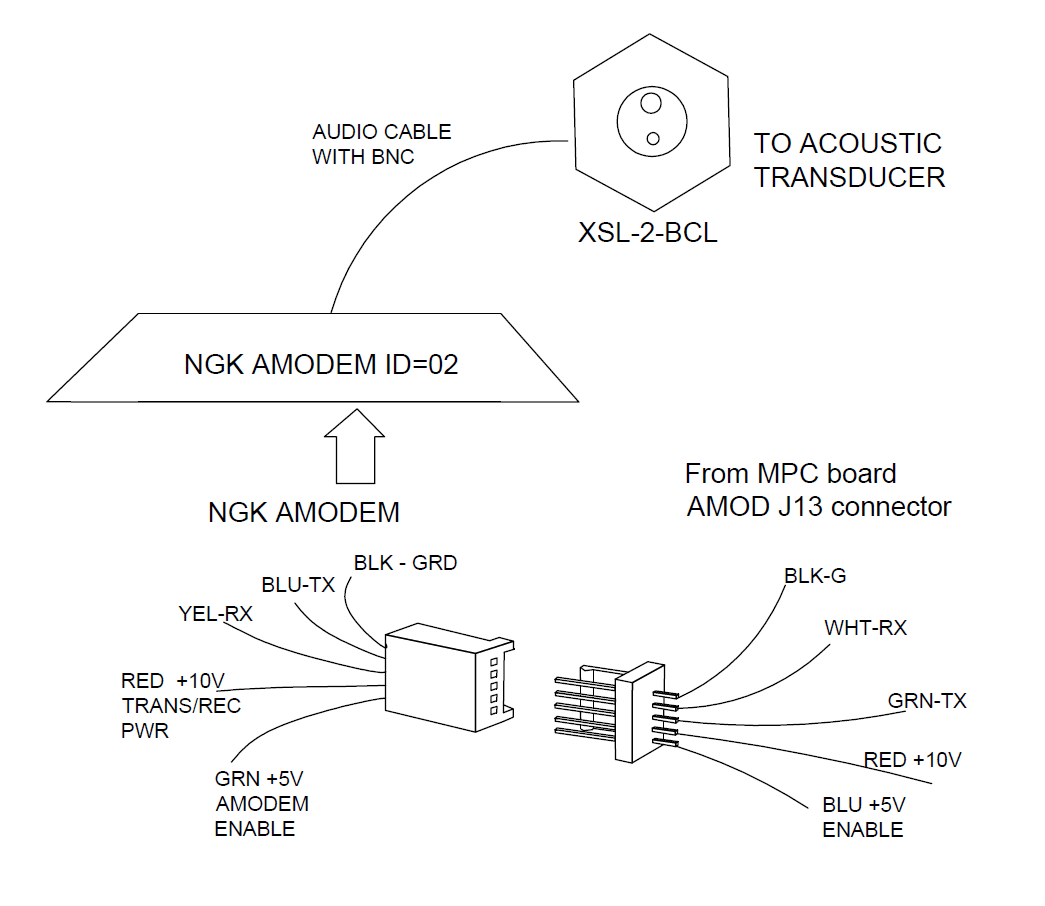


Figure 8. Amodem connections

**e. SBE16Plus V2 (CTD with Fluorometer)**

CF2/MPC communicates and provides power (VBAT without a regulator) to the SeaBird CTD SBE16plus through the CTD COM (COM1B J8). SBE16 has also a fluorometer and a PAR sensors mounted on the buoy frame. It uses Impulse XSG-4-BCL-HP-SS (Male) connector to get power and serial interface. The pin assignments of 4-pin connector are: pin 1=ground, pin 2=TX, pin 3=RX, pin 4=Power. These wires are connected to CTD/SEAGLID port (J8) on MPC. SBE16Plus power is always left on. When it is not addressed for longer than 25 sec, it goes back to low-power mode. To enable the CTD COM port, set CF2 pin 22 high.

The power to the SBE16 stays on all the time (VBAT directly from the battery). By setting the CF2 pin 22 on and open the com port (RX=pin 32 with IRQ2, TX=pin 31), CF2 communicates with SBE16plus. The command to retrieve the CTD data is ‘ts’ with a carriage return. The response should be

16.7301, 0.00832, 0.243, 0.0098, 0.0106, 0.0495, 14 May 2017 23:18:20

Temp conductivity depth fluromtr PAR salinity date and time

Expect latency of approximately 2 sec to get response. It can repeat the measurement at 4 sec interval.

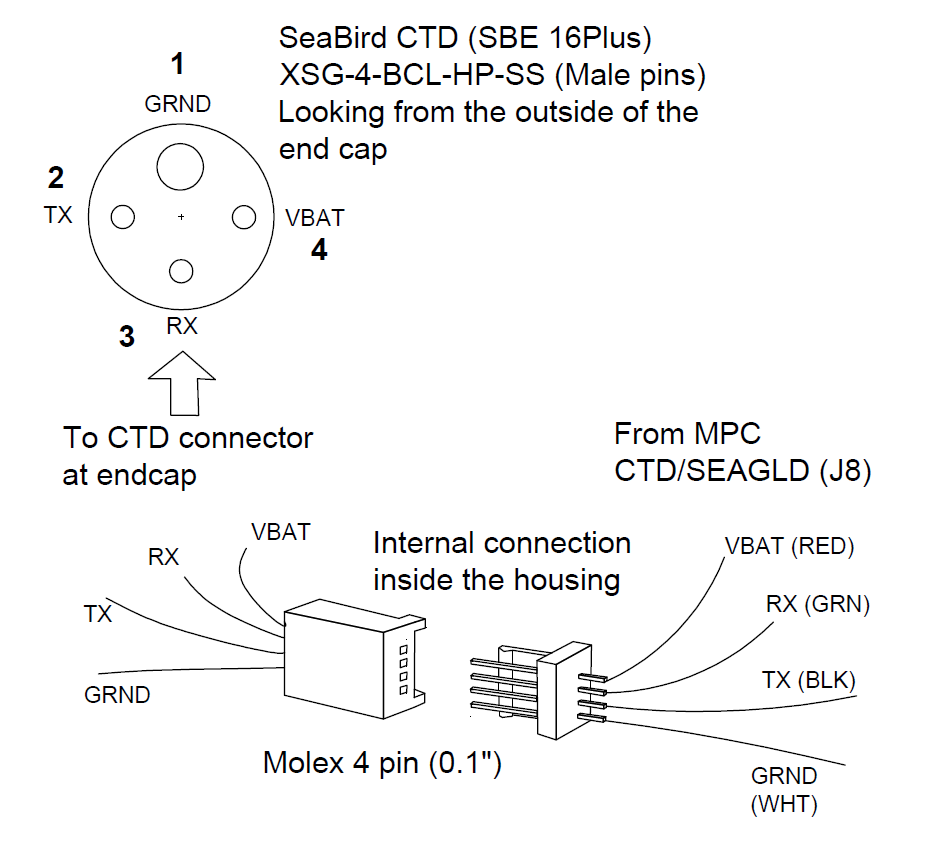


Figure 9. SBE16 connections

**f. Pre-Amp**

**EOS HM-1 pre-amp**

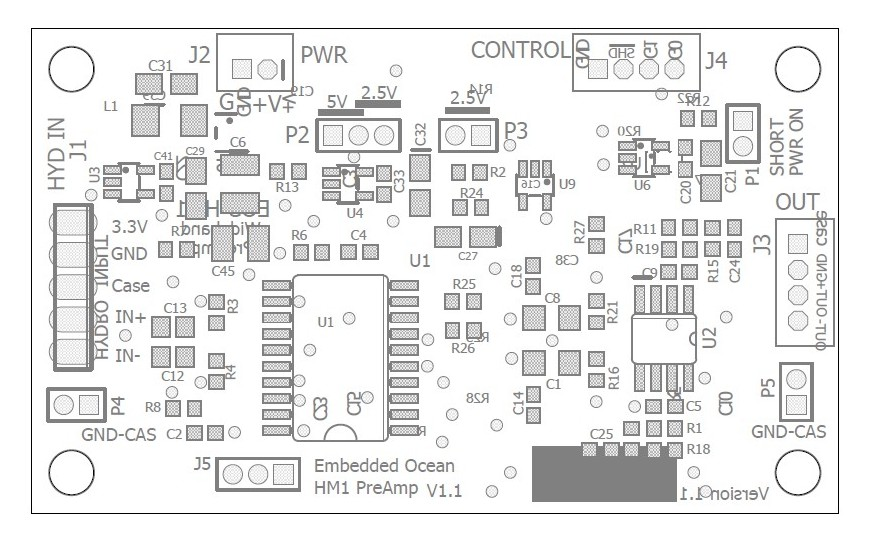


Figure 10. EOS pre-amp

MPC allows to mount up to four WISPR boards. Only one pre-amp is installed. Its signal output is multiplexed to an appropriate WISPR. The pre-amp draws ~4 mA at 12 V. The hydrophone (HIT92WB) signal outputs and power should be connected at J1.

|  |  |
| --- | --- |
| P1 | open |
| P2 (2.5/5V) | 5 V |
| P3 (2.5V) | short |
| P4 | short |
| J5 | open |

Jumper settings

|  |  |  |
| --- | --- | --- |
| G0 | G1 | Gain dB] |
| 0 | 0 | 0 |
| 1 | 0 | 6 |
| 0 | 1 | 12 |
| 1 | 1 | 18 |

Gain setting of pre-amp at J4

Pin 1 at J1 provides +3.3 V power to the HTI92WB’s built-in pre-amp which draws ~1.5 mA. The hydrophone outputs are differential with a shield as a ground. At P2 jumper, set the output range to 5 V by shorting across “5V”. At P3, set the mean voltage output “2.5 V” by shorting the 2-pin P3. Leaving P3 opened makes the mean 1.25V, which is reserved for output range of 2.5V applications (AUH).

P1 on the right hand corner is a manual power switch which is only used to turn on the power during lab tests or pre-amp gain calibration. It should be off-position (open) for shipment and operation. During normal operations, as the logging begins, the power is normally off until the logging begins. It is turned on only after the WISPR’s logging program starts and turns it on by making the ~~SHDN~~ pin of J4 high (3.3V to 5V). If you leave P1 pins shorted, its power stays on during shipment and shortens the battery life.

WISPR controls the gain in 4 steps with 6 dB increment by changing the state of G0 and G1 at J4 as shown in the table.

The two signal outputs at J3 are differential (0-5 V). So the total output between the two ADC channels is 10 V. For a single WISPR application (e.g., Seaglider or QUEphone) the outputs are connected directly to the A/D input of WISPR board (J9).

Note that WISPR’s ADC and reference voltage ICs are run by a separate power connector (J10). This is to isolate the digital noise. If no power is present at J10, ADC ship is not turned on and the WISPR data logging program stalls and no logging occurs,

P4 jumper shorts the shield ground to the power ground. For our HTI92WB hydrophone, the shield and power ground are tied together at the hydrophone, so P4 being shunt or open does not affect the signal quality.

**g. Analog Multiplxer (1:4 channel)**

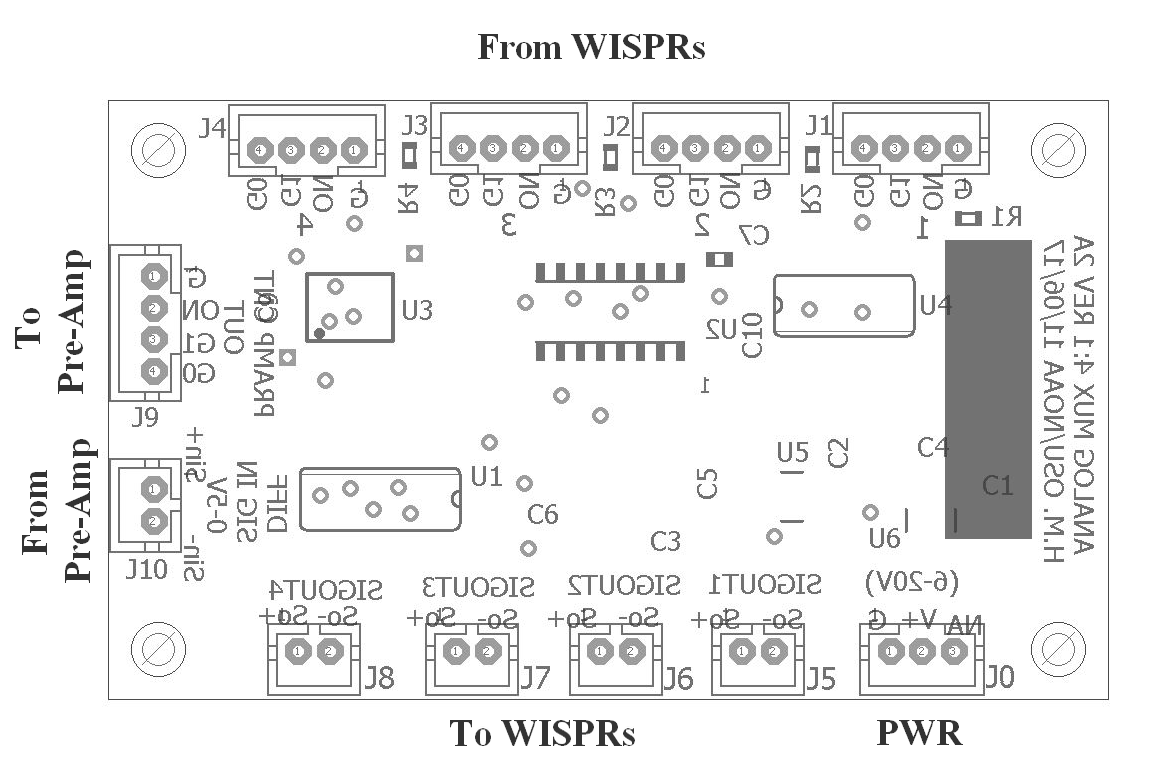


Figure 11. 4-channel analog multiplexer

The 4-channel multiplexer takes in a single pre-amp signal output at J10. It senses which WISPR is on by checking the PWR ON pin level at J1, J2, J3, J4 and routes the pre-amp output to the ADC of one of four WISPRS (J5, J6, J7, J8) accordingly. The rest of the WISPRs are in low-power sleep mode. When the WISPR is set to a sleep mode (by sending a 1-ms pulse to pin 4 of J3), it draws 2 mA. When the program is running, it draws approximately 75-100 mA.

As described in the previous section, MPC board can communicate and control up to four WISPR boards by multiplexing the pre-amp signal outputs to the ADC input of an appropriate WISPR. The powers of all the WISPR boards (J1 and J10) stay connected to the battery (draws ~2 mA). Also note that only one WISPR can run the logging program at any given time. Before a new logging cycle begins, CF2 switches the WISPR to a full power mode, and lets the logging program partially begin. First CF2 inquires the WISPR for the percentage of free data storage space. If the space is running short, during the next logging cycle, CF2 switches to the next WISPR. Logging stops when the data storage of the last WISPRs runs out.

**h. Hydrophone**

Model # : HTI-92-WB

Max pressure rating : 2,000 m

Sensitivity         : -165 dB re 1µPa/V (with internal pre-amplification)

Frequency response : 1-st order high pass at 20-Hz and no low pass filter at high frequency end

Power consumption   : < ~1.5mA total  
Supply Volt         : +3.3V  
Connector : Impulse connector 6-pin IE55-1206-CCP (straight)

Pin 1 : Negative signal output

Pin 2 : Positive signal output

Pin 3 : +3.3V DC

Pin 4 : Ground - 3.3V DC return (Ground+ Shield)

Pin 5 and 6 : NC

Mating connector IE55\_1206BCR on the top end cap and pin assignments.

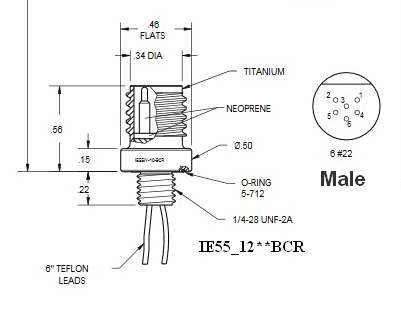


Figure 12. Hydrophone connector

1. **Battery**

**Alkaline**

11 D-cells in series and 9 in parallel. The output voltage is 16.5 V when it is new. Current capacity is approximately 135 AHr at 10 °C. Dimension 5.25” diameter 22” long. Weight 13.7 kg

**Lithium**

4 blocks of Lithium battery package. Lithium D-cells from Saft LSH20 (3.6 V 11AHr) are used. Each block is consisted of 4 D-cells in series (14.4 V) per column with 9 in parallel. 4 blocks in parallel have the current capacity of 396 AHr. Dimension 5.25” diameter 29” long. Weight 14.4 kg.

**j. End caps of the buoy module**

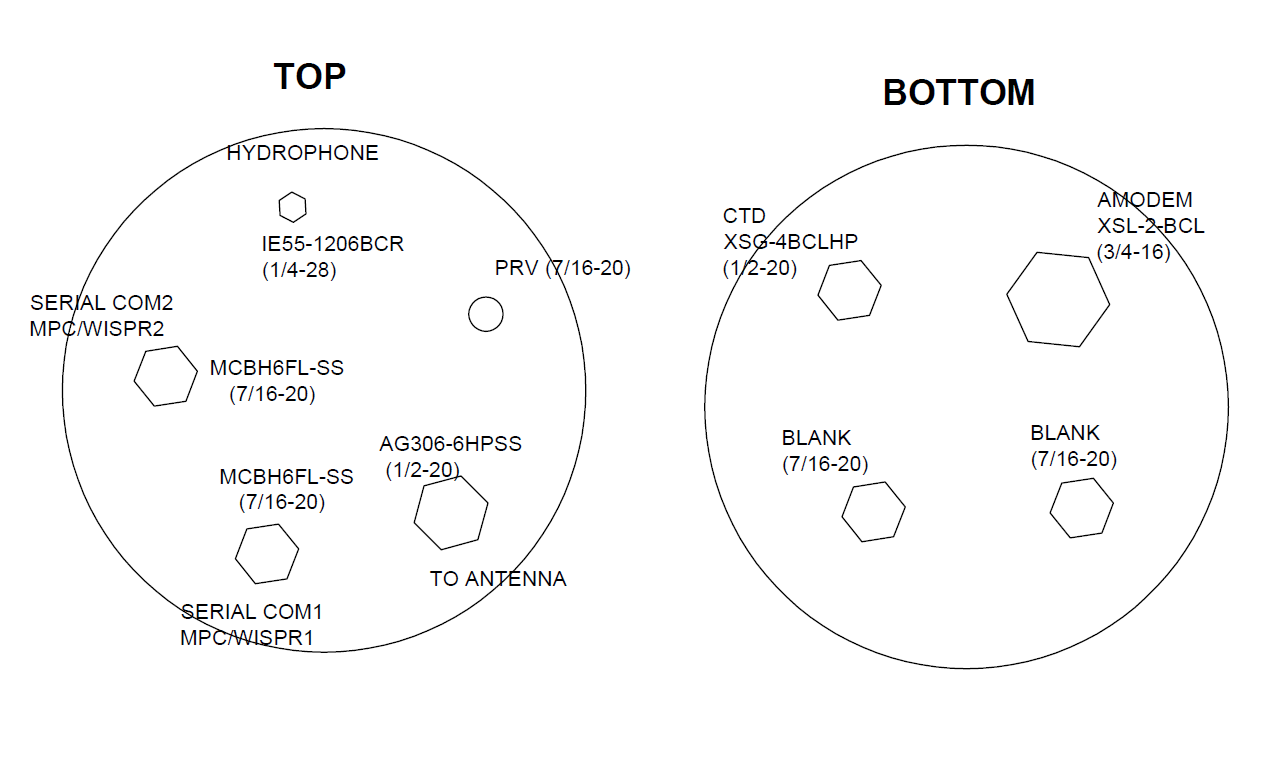
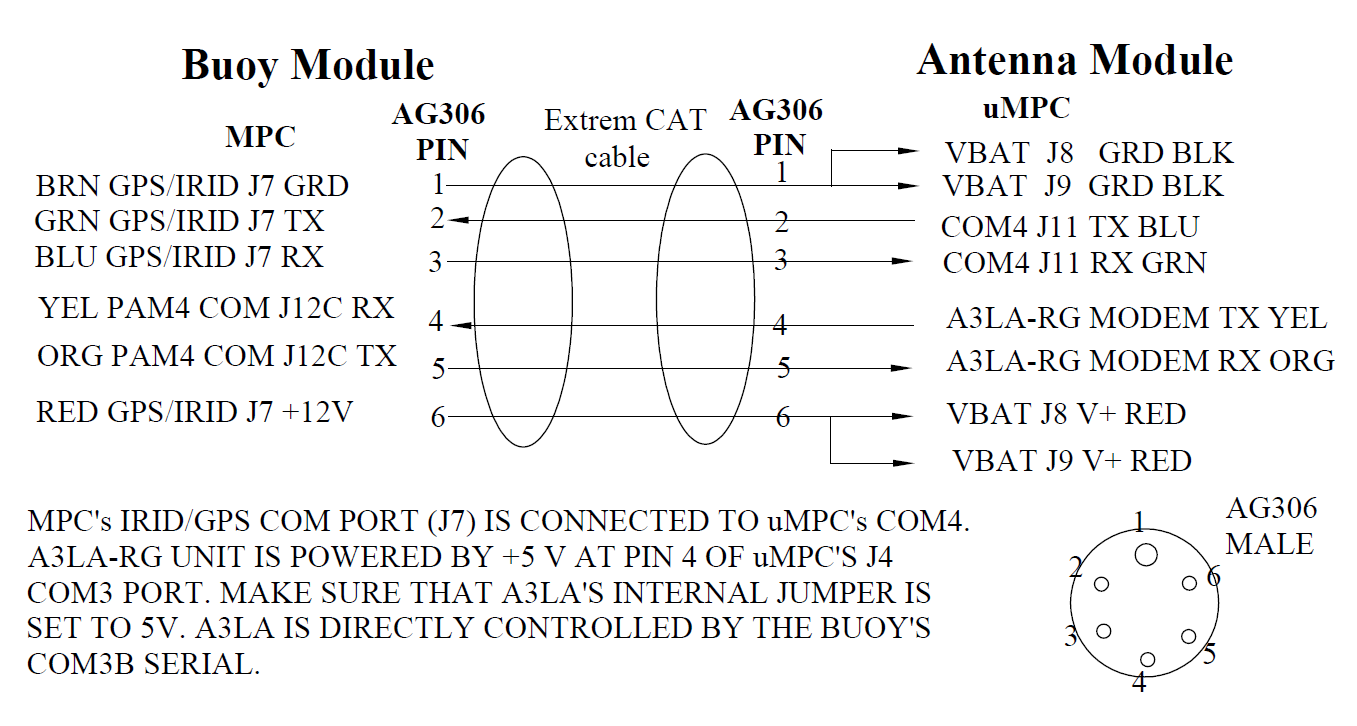


Figure 13. End caps

**k. Cable between buoy and antenna modules. Underwater connectors, pin assignments and wirings.**

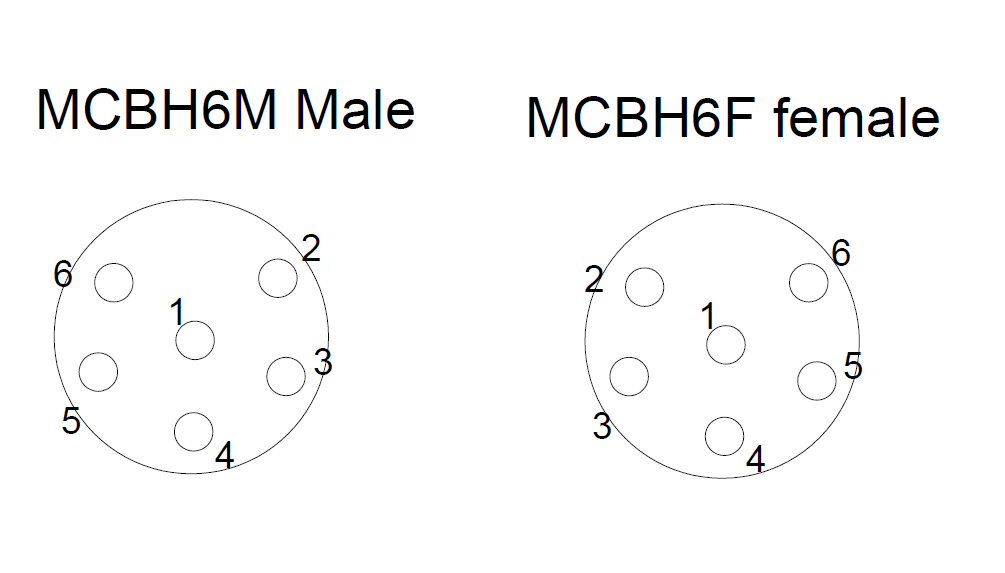


Figuer 14. Underwater cable connections between buoy and antenna

**l. Serial console com ports at the end cap**

There are two MCBH6 6-pin connector at the top end cap to communicate with CF2 and WISPR. One for the WISPR1 and CF2 and the other is for WISPR2 and CF2. WISPR communicaiton speed is 115200 and CF2 is 9600.

**MCBH6 connectors**



|  |  |
| --- | --- |
| Pin | Description |
| *1* | *Ground* |
| 2 | CF2 console (COM0) TX 9600 |
| 3 | CF2 console (COM0) RX 9600 |
| 4 | Ground |
| 5 | WISPR Console (COM0) RX 115200 |
| 6 | WISPT Console (COM0) TX 115200 |

1. **Antenna Module**

MPC converts the battery voltage to 12-V and sends it to the antenna by setting its CF2’s pin 23. Communication is enabled by setting its pin 22. When the buoy turns on the antenna power, it receives 12-V power to µMPC at VBATin connectors. It converts the 12 V to 5V for A3LA Iridium/GPS modem and 10 V for SBE39plus. Use the two VBATin power connectors to handle a large current to operate A3LA.

For a general serial communication, on MPC side it uses the IRID/GPS port (COM1A J7) to communicate with CF2 in the antenna. For the Iridium/GPS communication, MPC uses PAM4 port (COM3B) to talk to A3LA-RG modem directly.

Antenna module’s CF2/µMPC is powered by the buoy and communicates with buoy by COM4 (J1. Inside the antenna housing, there are 1) antenna switch (model 401-420 832 from Dow-Key Microwave, 2) A3LA Iridium/GPS modem from NAL, and 3) SBE39Plus temp and depth sensor from SeaBird. It also controls the antenna switch.

**µMPC board**

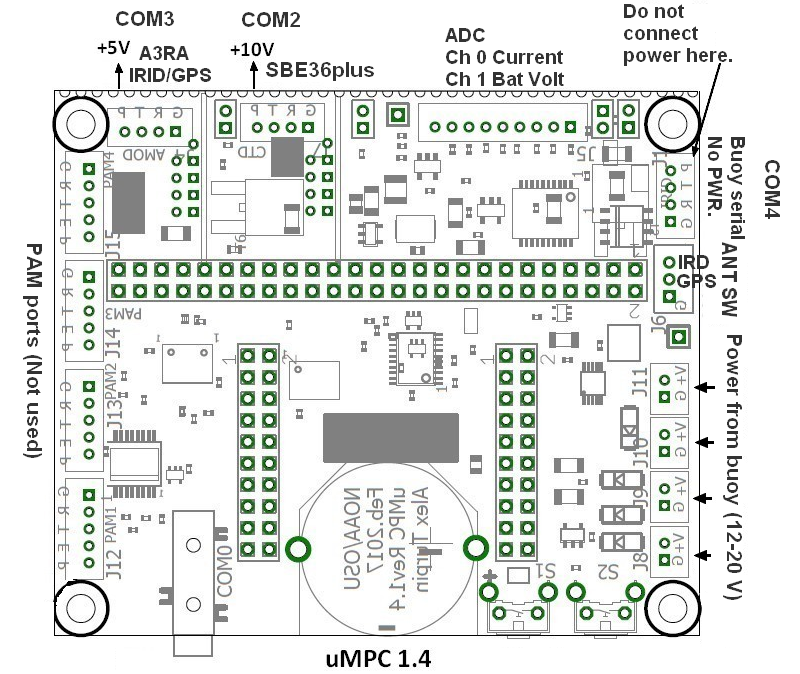


Figure 15. uMPC

|  |  |  |
| --- | --- | --- |
| uMPC | Connected to | Connections |
| COM2 | SBE39Plus | RX, TX, G and +10V |
| COM3 | Power to A3LA-RG IRIDGPS (no RX TX) | +5 V power only |
| COM4 | Buoy serial line (RX & TX only) | RX, TX and ground. No PWR connected here. |
| Ant SW | Antenna switch | 401-420 832 Dow-Key |
| PWR | DC Power from buoy (2 ports min) | +12 V and ground (11-20 V) |
| IRDGPS serial from buoy | A3LA-RG IRIDGPS modem | Two lines RX and TX |

**A3LA-RG (Iridium GPS modem)**

Using the two wires in the Extreme Cat cable (green), the buoy communicates direclty with the A3LA-RG IRID/GPS modem in the antenna. MPC uses its PAM COM3B to communicate and sends the 12-V power from COM1A to the buoy’s the VBAT ports of the uMPC in the antenna.

**SBE39plus**

CF2 sends ‘ts’ with a carriage return. The reply should be

23.8875, -0.015, 09 Jun 2017, 17:29:58

<Executed/>

If SB39Plus has not been accessed for longer than 2 min, it goes to low-power sleep mode. If it is in a sleep mode, it receives only a partial command of ‘ts’ and sends the following error message

<ERROR type=’INVALID COMMAND’ msg=’Cmd not recognized/’>

<Executed/>

You will have to send ‘ts CR” again. It is safe to send a carriage return first if the idle is longer than 2 min, ignore the first <ERROR response, then send ‘ts’ again.

**Antenna Switch (401-420 832 from Dow-Key)**

Send short pulse (maximum 150 ms) to pin 1 (/DS) to flip the relay of the antenna switch.

**Acoustic release**

**Model SMC (NGK)**

Max range 1500 m

TX freq 20.480 kHz and 19.692 kHz

Receive freq 19.948 kHz and 19.200 kHz

Max range 1500 m

Depth rating 500 m

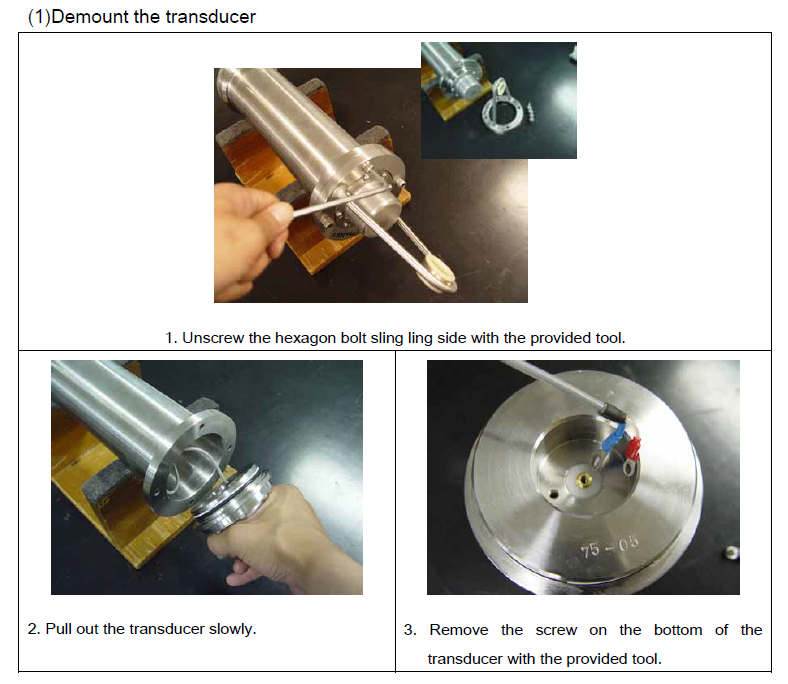
Max load 400 kg

Weight 5 kg in water

Battery For TX Alkaline 12 V (1.5 V 3A 8 in series, Panasonic LR6EJ)

Receive 7.2 V Lithium C two parallel (two 3.6 V C in series Tadiran TL-5920)

There is no external switch to power on/off. Open the pressure housing in order to connect or disconnect the two battery power plugs. See the pictures below.





**MPC board (Rev 4, March 22, 2018) and CF2 pin assignments for the buoy electronics system**

**Table 1-1. Connectors**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| J1-J4 | J5 | J6 | J7 | J8 | J9 | J10 | J11 | J12 | J13 | J14 |
| VBAT | **ADC inputs** | **ANT-SW** | **COM1A**  **GPSIRID** | **COM1B**  **CTD** | **COM2A**  **PAM1** | **COM2B**  **PAM2** | **COM3A**  **PAM3** | **COM3B**  **PAM4** | **COM4**  **AMOD** | **DIFAR**  **PWR** |

**PAM is WISPR board. Up to four WISPRs can be controlled.**

**Table 1-2. CF2 TPU and DIO assignments**

|  |  |  |
| --- | --- | --- |
| TPU | CF2 pin | WCNT Function Descriptions |
| 1 | 22 | Enable COM1 for GPS-IRID or SB16 CTD-SEAGLID. When ON, 12-V is send to antenna. (Originally 5 V) |
| 2 | 23 | Select COM1A (IRIDGPS) or COM1B (SB16 CTD-SEAGLID) |
| 3 | 24 | Select COM3 for PAM3 & PAM4 |
| 4 | 25 | Select PAM3 or PAM4 |
| 5 | 26 | H-P Sensor On/OFF (Originally for DiFAR PWR) |
| 6 | 27 | COM2&3 TX (For PAM1,2,3 &4) |
| 7 | 28 | COM2&3 RX (For PAM1,2,3&4) also IRQ5 |
| 8 | 29 | Select COM2 for PAM1&PAM2 |
| 9 | 30 | Select PAM1 or PAM2 |
| 10 | 31 | COM1 TX (GPS-IRID/CTD-SEAGLID) |
| 11 | 32 | COM1 RX (GPS-IRID/CTD-SEAGLID) connected to IRQ2 |
| 12 | 33 | COM4 RX (AMODEM) connected to IRQ3 |
| 13 | 34 | 0:Program Interrupt Enable |
| 14 | 35 | COM4 TX (AMODEM)\*\*\* |
| 15 | 37 | PAM PWR ON (send 1 ms pulse)\*\* |
| DS | 1 | Antenna SW (IRID/GPS)\*\* (Not used for LARA ) |
| PCS0 | 21 | AMODEM ON/OFF\*\* |
| PCS1 | 19 | AD VREF SHUTDOWN\*\* |
| MODCK | 42 | PAM PWR OFF (send 1 ms pulse) |

\*\*In the EEPROM environment, set sys.qpbc=32 in PicoDOS to enable non-TPU pins as GPIOs

\*\*\*You must set PIORead(48) to assert RTS signal to COM4

**Table 1-3. Sensor COM assignments**

|  |  |  |
| --- | --- | --- |
| Sensors | PWR | Com |
| Antenna module | Pin 22 12V | COM 1A (variable) |
| IRID/GPS in antenna | PWR on/off by CF2 in antenna | COM 1B (variable) |
| \*PAM1 | OFF or ON | COM2A (9600) |
| \*PAM2 | OFF or ON | COM2B (9600) |
| \*PAM3 | OFF or ON | COM3A (9600) |
| SBE16 CTD | PWR on all the time | COM3B |
| AMODEM | OFF or ON | COM4 (4800) |
| Hydrophone& PreAmp | PWR controlled by PAM DSP GPIO | Output connects to all PAMs inputs |
| H-P sensor | Pin 26 | Analog out ->ADC 2 & 3 ch |

**\*Only one PAM can be ON and communicated at any time.**

**Table 1-4. IRQ Table**

|  |  |  |
| --- | --- | --- |
| IRQ | Pin | Functions |
| 2 | **41** | **COM1 GPS-IRID/CTD/Fluoro interrupt** |
| 3 | **50** | **AMODEM interrupt** |
| 5 | **39** | **COM2&3 PAM 1,2,3&4 int (also default interrupt for Rev 2)** |
| 4 | **Reserved** | **Console Com (COM0) interrupt** |
| 7 | **40** | **Reserved for C: access QSPIII data logging** |

**Table 1-5. COM 1 Channel Select (GPS/IRID or SBE16 CTD-SEAGLID. SBE16 CTD’s power is left ON)**

|  |  |  |  |
| --- | --- | --- | --- |
| TPU 1 (22) Antenna PWR | TPU 2 (23)  Select COM | COM1A or  COM1B | Antenna PWR  +12V |
| 0 | **0** | **All COM OFF** | **OFF** |
| 1 | **0** | **COM1B J8 A3LA** | **ON** |
| 1 | **1** | **COM1A J7-To Buoy** | **ON** |

**Table 1-6. COM 2 Chan Select (PAM1&2) A pulse to pin 37 to on, pin 42 to shut off PAM**

|  |  |  |  |
| --- | --- | --- | --- |
| TPU 8 (29 COM2 PWR ON) | TPU 9 (30) | COM2A/B | PWR |
| 0 | **X** | **COM OFF** | **PAM1&2 OFF** |
| 1 | **0** | **PAM1 J9C** | **PAM1 ON** |
| 1 | **1** | **PAM2 J10C** | **PAM2 ON** |

**Table 1-7. COM 3 Chan Select (PAM3&4) A pulse to pin 37 to on, pin 42 to shut off PAM**

|  |  |  |  |
| --- | --- | --- | --- |
| TPU 3 (24 COM3 PWR ON) | TPU 4 (25) | COM3A/B | PWR |
| 0 | **x** | **COM OFF** | **PAM3&4 OFF** |
| 1 | **0** | **PAM3 J11C** | **PAM3** |
| 1 | **1** | **SB16 J12C** | **PAM4** |

**Table 1-8. COM4 AMODEM**

|  |  |  |
| --- | --- | --- |
| PCS0 (Pin 21) | COM4 | PWR |
| 0 | **OFF** | **OFF** |
| 1 | **AMODEM** | **ON** |

**Table 1-9. H-P sensor power (DiFAR)**

|  |  |
| --- | --- |
| TPU5 (pin 26) | H-P sensor PWR |
| 0 | **OFF** |
| 1 | **ON** |

**Table 1-10. Program interrupt**

|  |  |
| --- | --- |
| TPU13 (34) |  |
| 0 | **IRQ5 for Program Interrupt Enable** |
| 1 | **IRQ5 for PAM COM RX interrupt** |

**Table 1-11. Antenna Switch (not used)**

|  |  |
| --- | --- |
| DS (Pin 1) | Power On (VBat voltage) |
| 0 | **GPS (200mA Latch current)** |
| 1 | **IRID (200mA Latch current)** |

**Table 1-12. Power Specs and baud rate of each sensor**

|  |  |  |  |
| --- | --- | --- | --- |
| Device | Input Voltage [V] | Current [mA] | RS232 |
| IRID/GPS AL3A | 5 V | 2.2AMax/200mA Typ | 9600 |
| SBE16 CTD/Fluorometer | 9-28 V  Always ON | 9.1mA/30uA pump 25mA  250mA | variable |
| NGK AMODEM | 10-18V Xmit & 5-10V receiver board powers are tied together. MPC sends 10 V 2A.  Enable (5V) to wake up. 5V is always ON. Toggling 10-V to turn power ON/OFF/ | 1500mA(12V)  30mA(7.2V)  0.5mA sleep | 4800 |
| PAM/Pre-Amp | 6-20 V | 75mA/2mA  Separate battery for DSP&VRef&Pre-Amp | 9600 &  115200 |

**µMPC board and CF2 Pin assignments**

**in antenna module (March 20, 2018)**

**Table 2-1. Connectors**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| J1 | J4 | J5 | J6 | J7 | J8-J11 | J12 | J13 | J14 | J15 |
| To buoy COM4 & PWR | **IRID/GPS**  **COM3** | **ADC inputs** | **ANT**  **SW** | **SBE39PLUS**  **COM2** | **VBAT**  **Use at least two** | **PAM1**  **COM1** | **PAM2**  **COM1** | **PAM3**  **COM1** | **PAM4**  **COM1** |

**Table 2-2. CF2 TPU and DIO assignments**

|  |  |  |  |
| --- | --- | --- | --- |
| TPU | CF2 pin | IN/OUT | Function Descriptions |
| /DS | 1 | OUT | Antenna SW. A short pulse ~100ms (IRID/GPS)\*\* |
| PCS1 | 19 | OUT | AD VREF SHUTDOWN\*\*6 |
| PCS0 | 21 | OUT | IRID/GPS module power ON/OFF (continuous) |
| 1 | 22 | OUT | 1= Power on COM4 VBAT out (do not connect pwr for this revision) |
| 2 | 23 | OUT | 1= Power on SBE 39Plus (COM2) 10 V out |
| 3 | 24 | OUT | Enable/disable all PAM |
| 4 | 25 | OUT | TXout COM4 (Buoy module) |
| 5 | 26 | IN | RXin COM4 ( tied to IRQ5) |
| 6 | 27 | OUT | PAM TXout |
| 7 | 28 | IN | PAM RXin (no IRQ) |
| 8 | 29 | OUT | Select PAM |
| 9 | 30 | OUT | Select PAM |
| 10 | 31 | OUT | TD SBE39Plus TX out (COM2) |
| 11 | 32 | IN | TD SBE 39Plus RX in (IRQ2) COM2 |
| 12 | 33 | IN | IRID/GPS COM3 CMOS logic RX in (connected to pin 50 and IRQ3) |
| 13 | 34 | IN | Not used |
| 14 | 35 | OUT | IRID/GPS COM3 CMOS logic TX out (connected to pin 48) |
| 15 | 37 | OUT | PAM1 and 2 enable signal (send 1 ms pulse) |
| MODCK | 42 | OUT | PAM3 and 4 enable signal (send 1 ms pulse)\*\* |
| RSRTS | 47 | OUT | IRID/GPS (COM3) UART level TX out (+/- 5V) |
| /RTS | 48 | IN | CMOS COM3 logic in. UART output is 47 |
| RSCTS | 49 | IN | IRID/GPS COM3 UART level RX in (+/- 5V) |
| IRQ3/CTS | 50 | OUT | CMOS also IRQ3 (IRID/GPS COM3) from pin 49 |

\*\*Set sys.qpbc=32 in PicoDOS to enable these pins as DIOs

\*\*\* PIORead(48) to assert RTS signal

**Table 2-3. PAM COM1 assignments ~~(IRQ5 interrupt) Only PAM is used to communicate with buoy electronics.~~ Power off pin is missing. In the future, TPU13 can be used for power-off signal for all PAMs.**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Sensors | PWR | TPU3  (24) | TPU8  (29) | TPU9  (30) | TPU15  (36)\*\* | MODCK (42) | TPU6 | TPU7 |
| No PAM | VBAT | 0 | x | x | x | x | x | x |
| \*PAM1 (J12) | VBAT | 1 | 0 | 0 | pulse | x | TX | RX |
| PAM2 (J13) | VBAT | 1 | 1 | 0 | pulse | x | TX | RX |
| PAM3 (J14) | VBAT | 1 | 0 | 1 | x | pulse | TX | RX |
| PAM4 (J15) | VBAT | 1 | 1 | 1 | x | pulse | TX | RX |

**~~\*PAM1 COM should be enabled all the time. No need to send a pulse to COM port.~~**

**Table 2-4. ~~PAM1 Channel Select (to communicate with buoy electronics, stays on all time)~~**

|  |  |  |  |
| --- | --- | --- | --- |
| TPU 3 (24)  PWR ON | TPU 2  (23) | TPU 8  (24) | TPU 9  (25) |
| 0 | **x** | **x** | **x** |
| 1 | **1** | **0** | **0** |
| 1 | **1** | **0** | **0** |

**~~Must to set TPU2 and TPU 9 to choose PAM1 before powering up.~~**

**Do not use PAM com (COM1) for a long wire serial com, because it is multiplexed by MAX399 (analog). It works for a short distance, but not for long wires. To talk over a long range such as a buoy, use COM4 instead. HM, 2018**

**Table 2-5. SBE39Plus (COM2)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TPU 2  (pin 23)  PWR SW | PWR  (10 V) | PCS0  (pin 21)  Logic only | TPU10  (pin 31) | TPU11  (pin 32)  IRQ2 |
| 0 | **OFF** | **x** |  |  |
| 1 | **ON** | **x** | **TX** | **RX** |
| 1 | **ON** | **x** | **TX** | **RX** |

**Set TPU 1=0 to disable COM4.**

**Table 2-6. A3LA IRID/GPS (COM3). Only PCS0 pin 21 is used for power. RX&TX not connected. A3LA’s directly talks to the buoy COM1B (J8 CTD/SEAGLID).**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| PCS0\* (Pin 21)  PWR SW | PWR  (5 V) | TPU 2  (pin 23) | TPU 3  (pin 24) | RSRTS  (Pin 47) | RSCTS  (Pin 49)  IRQ3 |
| 0 | **OFF** |  |  |  |  |
| 1 | **ON** | **x** | **x** | **TX** | **RX** |
| 1 | **ON** | **x** | **x** | **TX** | **RX** |
| 1 | **ON** | **x** | **x** | **TX** | **RX** |

**Table 2-7. Logic status of three COMs**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| TPU 3  (Pin 24) | PCS0  (pin 21) | TPU 2  (pin 23) | SBE 39  (COM2) | ~~IRID/GPS~~  Not used (COM3) | PAM  (COM1)  Not used | Buoy  (COM4) |
| 0 | **0** | **0** | **N** | **~~N~~** | **N** | **Y** |
| 0 | **0** | **1** | **Y** | **~~N~~** | **N** | **Y** |
| 0 | **1** | **0** | **N** | **~~Y~~** | **N** | **Y** |
| 0 | **1** | **1** | **Y** | **~~Y~~** | **N** | **Y** |
| 1 | **0** | **0** | **N** | **~~N~~** | **Y** | **Y** |
| 1 | **0** | **1** | **Y** | **~~N~~** | **Y** | **Y** |
| 1 | **1** | **0** | **N** | **~~Y~~** | **Y** | **Y** |
| 1 | **1** | **1** | **Y** | **~~Y~~** | **Y** | **Y** |

**Keep Pin 24 low to disable COM1 (PAM) to save the power.**

**Table 2-8. IRQ assignments**

|  |  |  |
| --- | --- | --- |
| IRQ | Pin | Functions |
| 2 | **41** | **COM2 SBE 39plus interrupt** |
| 3 | **50** | **COM3 IRID/GPS interrupt** |
| 5 | **39** | **COM4 Buoy Elec (also ext interrupt)** |
| 4 | **Reserved** | **Console Com (COM0) interrupt** |
| 7 | **40** | **Reserved for C: access QSPIII data logging** |

**Table 2-9. Program interrupt**

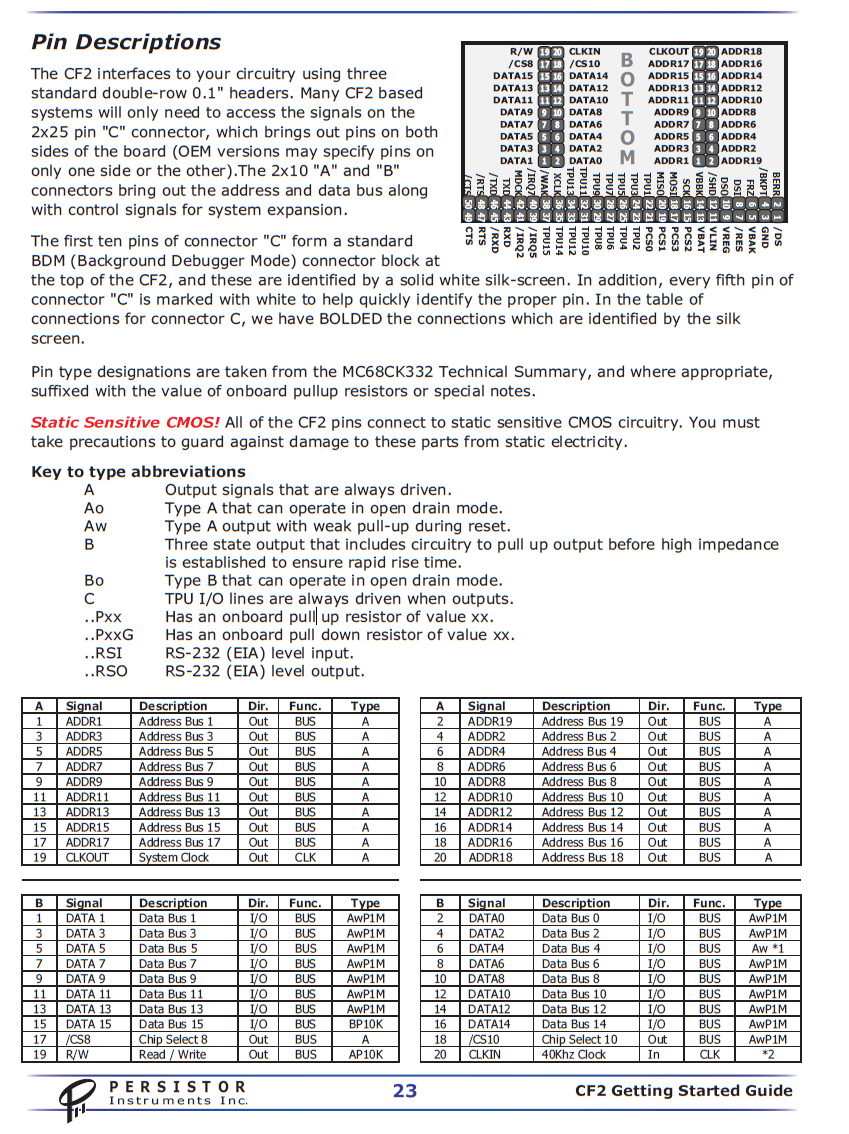
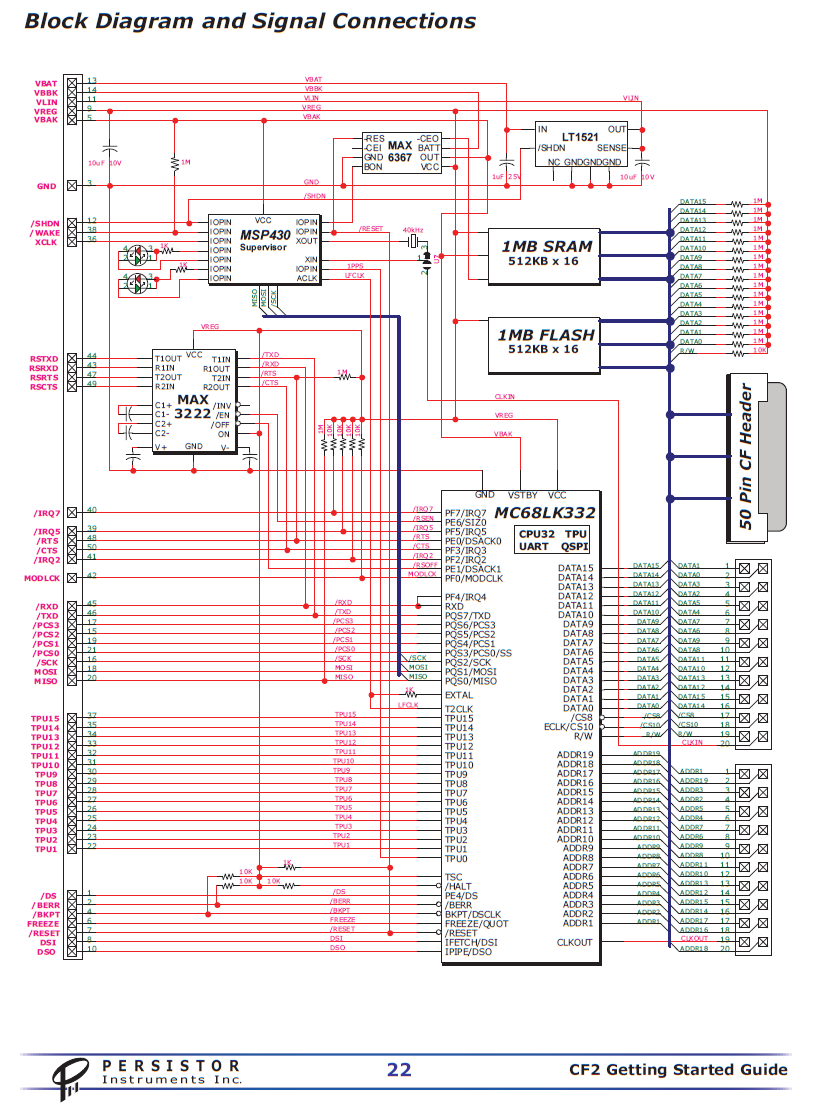
|  |  |
| --- | --- |
| TPU7 (pin 28) |  |
| 0 | **IRQ5 for Program Interrupt Enable** |
| 1 | **IRQ5 for PAM COM RX interrupt** |

**Table 2-10. Antenna Switch**

|  |  |
| --- | --- |
| /DS (Pin 1) | Send a short pulse (~100 ms) |
| 0 | **GPS (200mA Latch current)** |
| 1 | **IRID (200mA Latch current)** |

**Table 2-11. Power Specs and baud rate of each sensor**

|  |  |  |  |
| --- | --- | --- | --- |
| Device | Input Voltage [V] | Current [mA] | RS232 |
| IRID/GPS AL3A | 5 | 2.2AMax/200mA Typ | 9600 |
| SBE 39plus TD | 8-15  Power ON as needed (10 V nominal) | During TX 6 mA  IQ 25uA | 9600 |



**CF2 schematics**

