EM-APEX

Electromagnetic Autonomous Profiling Explorer

Operator's Manual

by Jim Carlson, John Dunlap, James Girton, and Thomas Sanford

March 2006

Applied Physics Laboratory University of Washington
1013 NE 40th Street Seattle, Washington 98105-6698

Contract N00014-03-C-0242
Subcontract Webb Research Corporation

Acknowledgments

The authors thank Janet Olsonbaker and Brian Rasmussen of APL-UW for their extensive and essential contribution to the organization and production of this operator's manual.

The Applied Physics Laboratory, University of Washington (APL-UW) thanks Webb Research Corporation (WRC) for their assistance. This work was funded by WRC as a subcontract on ONR SBIR contract N00014-03-C-0242.

Table of Contents

1. Introduction

The Electromagnetic Autonomous Profiling Explorer (EM-APEX) is the result of a collaboration between Webb Research Corp (WRC) and the Applied Physics Laboratory of the University of Washington (APL-UW). WRC is the manufacturer of the APEX autonomous profiling float (Fig. 1) and APL-UW is the developer of proven electromagnetic current measurement instrumentation that obtains velocity through motional induction (Fig. 2). The EM-APEX is a standard WRC APEX profiling float with an added APL-UW subsystem for measuring water velocity, specifically motionally-induced electric fields generated by the ocean currents moving through the vertical component of the Earth's magnetic field. These measurements can be (and have been) used to characterize internal wave properties, upper-ocean dynamics, and patterns of geostrophic shear, even under hurricane conditions.

Temperature (T) and salinity (S) observations are obtained by the Sea Bird Electronics SBE-41 CTD designed for the profiling floats of the global Argo array (http://www.argo.ucsd.edu). Electrodes on a right cylindrical shell, surrounding the lower half of the float, sense the motionally-induced voltages. The voltages are amplified, digitized, and stored within the float. Other components of the added subsystem are a magnetic compass and tilt sensors. Float position is determined by the GPS system when the float surfaces. The T, S, V (velocity), position, and other observations are processed on board the float and transmitted over the Iridium global satellite communication system.

This manual presents the float design and construction specifications, plus basic instructions for the operational use of the EM-APEX.

Fig. 1. APEX, the Webb Argo Float

Fig. 2. EM-APEX with developers, senior engineers Jim Carlson (left) and John Dunlap (right), prior to Puget Sound testing. Modifications to the APEX float include an internal electromagnetic subsystem, external PVC shell with electrodes, small turbine blades (sometimes referred to as vanes or fins) for instrument rotation, and Iridium/GPS patch antenna (shortened for air deployment on the float shown). The orange flag was used during testing only.

APEX or Webb Argo float specifications	
Maximum depth	2000 m
Endurance	4 years, 150 ascents
Mass	25 kg (EM subsystem adds 3 kg)
Dimensions	16.5 cm diameter x 127 cm long (not including antenna)

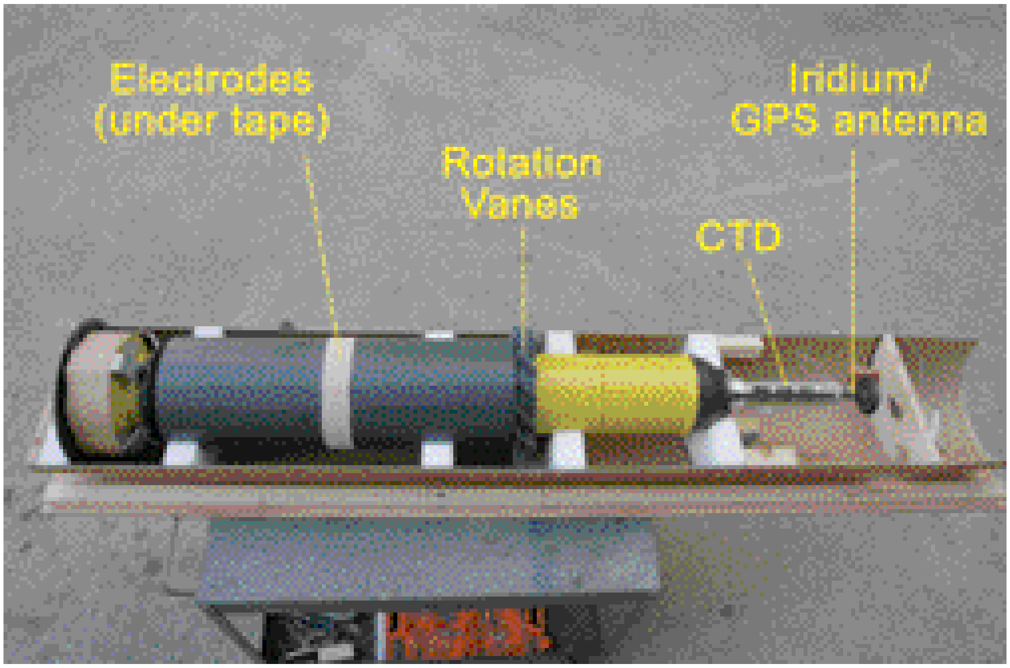

Fig. 3. The EM-APEX can be deployed from an airplane in its air-launch container. The top half (not shown) includes the top portion of the cardboard cylinder and parachute. The air-launch container disintegrates in seawater and is held together with nylon straps that are released by salt blocks that dissolve in about one hour after being immersed in seawater.

2. Motional Induction Theory

The velocity determination is based on the principles of motional induction that, in turn, have to be modified for a free drifting or free falling profiler such as the EM-APEX. The motion of seawater through the Earth's magnetic field produces electric currents and magnetic fields. The horizontal electric field as observed on a platform moving with the surrounding water is

$$\nabla_h \phi_a = -F_z\left(v(z) - \bar{v}^*\right) \times k - J^* / \sigma,$$

where $\nabla_h \phi_a$ is the apparent potential around a moving sensor, F_z is the vertical component of the Earths magnetic field, ϕ is electrical conductivity, v is local water velocity, $\bar{v}^*$ is a vertically integrated, conductivity-weighted ocean velocity, and J^* are nonlocal electric current density (typically negligible). Hence the measured ocean voltage gradient can be converted to a relative velocity profile:

$$v(z) - \bar{v}^* = k \times \nabla_h \phi_a / F_z$$

The important point is that only one term varies with depth. It is this term that provides the vertical distribution of current. The other terms represent an unknown (but knowable with an independent velocity measurement) depth-independent offset.

3. Design and Construction of EM-APEX

Science specifications for electric field measurements require velocity values with uncertainty < 1 cm/s. The electric subsystem must also operate on low power, permitting the float to function for up to five years. These requirements were met with custom low-noise preamplifiers and data acquisition system; the whole EM subsystem power requirement is under 70 mW when operating (i.e., when not in sleep mode).

External modifications

The exterior of the standard WRC APEX floats were modified in two essential ways. An exterior plastic skin was added, under which five Ag–AgCl electrodes in PVC housings were installed. Two pairs of electrodes were mounted under the shell to form two orthogonal, voltage-measuring axes. The fifth electrode is the reference electrode for the preamplifiers.

The second modification, external vanes, also called fins, were added to rotate the profiler as it descends/ascends through the water column (see Figs. 2 and 3). This rotation is needed primarily to remove the electrode drift from the voltage measurement and also to ensure that each electrode channel samples both components of velocity, in case of failure in one channel. A magnetic compass was added to allow conversion of the rotating electrode measurement into geographical coordinates.

A prototype Iridium/GPS antenna, shortened to accommodate air deployment, was mounted to the top cap in place of the Argo satellite antenna.

Internal modifications

The EM subsystem required relatively few internal modifications to the APEX: three additional circuit boards (EMA board, magnetometer/accelerometer board, amplifier board), batteries, and wiring. The air pump was relocated to provide a clear space to mount the additional electronics. The air bladder (about 1300 ml) provides buoyancy and stability while the EM-APEX is surfaced, lifting the antenna support out of the water. The bladder is filled via the air pump system.

During assembly, care was taken to de-magnetize any materials with hard iron characteristics. In place of the standard APF-8 controller and Système Argos PTT, the EM-APEX includes a new sub-chassis that incorporates (at the time of initial construction) the latest APF-9 controller, an Iridium L-Band Transceiver (LBT), a Garmin GPS receiver, and an RF (radio frequency) switch assembly to allow the antenna to be switched between the Iridium satellite phone and the GPS receiver. Note that the GPS and Iridium cannot be operated simultaneously.

GPS and Iridium

The Garmin GPS locates the EM-APEX float position while on the surface. If the GPS has been turned on recently, it will take about one minute to acquire a local position, otherwise it can take up to four minutes. The almanac, i.e., table of all satellite locations, has to be less than two weeks old, and it takes 12.5 minutes to download a new almanac.

Iridium Satellite LLC (www.iridium.com) provides global satellite voice and data over the oceans, airways, and polar regions. The Iridium constellation of 66 low-earth orbiting (LEO), cross-linked satellites operates as a fully meshed network. Users can configure EM-APEX connection times themselves to prevent excessive waiting on the surface in bad weather.

Vane ring, PVC shell, and electrodes

The APEX external damping ring was removed. A PVC-based assembly was mounted to the outside of the APEX pressure housing that incorporates the vane ring and the mounts for the five electrodes (see Figs. 2 and 3). The vane ring consists of sixteen blades or fins mounted in a compact assembly designed to rotate the APEX as quickly as possible given the dimensional constraints of the air deployment package. Ag–AgCl electrodes are placed in chambers filled with agar made with saltwater, which acts to buffer the electrodes from temperature and salinity changes. The EM electrodes are mounted under the PVC shell to present a smooth, cylindrical surface for the ocean electric currents to divert around the body and to minimize the effect of voltages induced by water flow around the hull. The electrodes are attached to a wiring harness that enters the pressure housing through an underwater penetrator in the bottom end cap. The wiring harness utilizes extra long wires that allow disassembly of the EM-APEX. When assembled, the extra wires are wrapped around the bottom cap.

Preamplifiers

The two orthogonal electrode pairs are sampled by low-power, low-noise preamplifiers with 24-bit resolution at 20 samples per second (SPS). The amplifiers have a gain of 500 and are DC-coupled providing a full-scale input of ±5 mV. The electrodes are spaced 0.215 m apart, giving an electric field resolution of 1.4 nV/m, corresponding to a velocity resolution of 0.014 cm/s in a vertical geomagnetic field of 1×10^4 nT. This is at least an order of magnitude below the dominant noise contributions to the velocity measurement.

Magnetometer and tilt sensor

Included are three channels of orthogonal acceleration and three channels of orthogonal magnetic field. Two of the acceleration channels (tilt) have full-scale ranges of ±0.25 g. The third (vertical) acceleration channel has a full-scale range of ±2 g. The three magnetic field channels have a full-scale range of $\pm2\times10^{-4}$ T and are used in conjunction with the tilt sensors to locate magnetic north, as well as to estimate the total field strength. To save energy, all six orientation channels are powered off between bursts: every second they are powered up for 70 ms, sampled 11 times in 18 ms at 10-bit resolution, and then powered down.

Serial interface and batteries

The EM-subsystem is isolated from the APEX float electronics with an optically isolated serial interface and an independent and isolated set of batteries. The subsystem is optimized for low-power operation with many of the components "power-switched," i.e., the power is turned off when it is not needed. There are two battery packs: a unipolar pack consisting of five AA cells powers the microcontroller, accelerometers, and magnetometers, and a bipolar pack consisting of six AA cells powers the low noise amplifiers. When active the average power drawn by the EM subsystem is about 56 mW. These AA-cell battery packs contain about 26.4 W-h that provide a total active profiling time of 350 hours.

APEX

The battery packs consist of two stacks of alkaline D-cells of 10 cells each and one stack of C-cells for a total of 30 cells. The energy in two stacks of 10 alkaline D-cells (14 Ah) and one stack of 10 alkaline C-cells (7 Ah), using 1.0 V/cell, is about (14*10 + 14*10 + 7*10) * 1.0 * 3600 = 1.26 MJ (megajoules).

EM subsystem

The EM subsystem has a separate set of batteries consisting of 11 alkaline AA-cells. The energy in 11 alkaline AA-cells (2.45 Ah) is about 2.45 * 11 * 1.0 * 3600 = 97 kJ (kilojoules), or about 0.1 MJ.

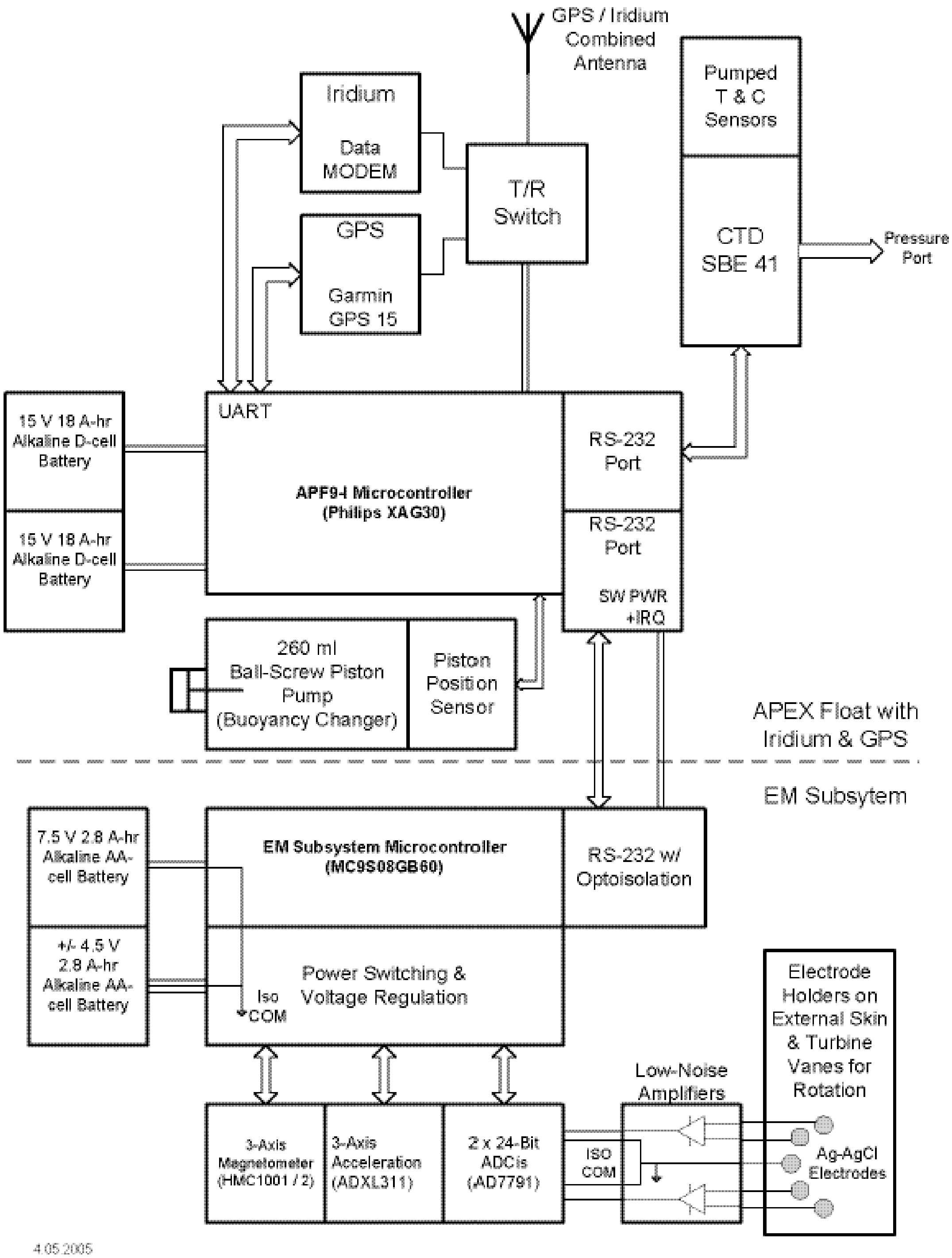

Fig. 4. Block diagram of the EM subsystem for the EM-APEX float

Fig. 5. Circuit boards in EM subsystem of EM-APEX

Microprocessors

The control functions and firmware in the EM-APEX are split between two microprocessors:

- A Motorola Freescale MC9S08GB60 on the EMA board samples the electric field data (20 Hz) as well as magnetic and tilt data (11 sample burst every second). It also computes 1-s averages of the sampled data. Up to 25 s of this data is stored in the EMA micro RAM to allow unhurried requests from the APF9 controller. If there are no requests for data in 30 s, the EMA board shuts down to 20 microamps. This allows for multi-year missions in which the EM subsystem is periodically awakened and velocity profiles are obtained.
- A Philips XAG30 microprocessor on the APF9 board is responsible for all of the APEX operations. Firmware was obtained from Dana Swift and Steve Riser of the University of Washington School of Oceanography and modified as detailed below for the EM-APEX.

The 8-bit Motorola microprocessor in the EM subsystem was chosen because of its very low power as it runs continuously during EF data sampling. However, because it has only 4 K of RAM (together with a 60-K Flash for code) and its floating point arithmetic is much slower than the 16-bit XAG30, the EF sinusoidal fitting is done using the XAG30 microprocessor on the APF9 controller board.

New EM-APEX capabilities

EM-specific code in the APF9 computes sinusoidal least squares fits using 50 s of 1-s averages from the EMA board. The fits are overlapped 25 s, i.e., 50 percent (though both the fit length and the overlap are adjustable parameters—see Appendix A). The EF least squares fit coefficients, some EMA status information, CTD profiles, and GPS positions are sent to the home server via the Iridium connection each time the float surfaces. The fits and pressure data are combined after reception ashore to provide velocity profiles.

Besides this EM processing code added to the APF9, several new profiling mission capabilities were added to the standard APF9 APEX code base:

a. Commences mission upon reaching a set pressure (Prelude mode)

b. Samples while descending

c. Maintains vertical speed control while descending (Fast Profiling mode)

d. Samples continuously, storing multiple profiles, without parking or coming to the surface for telemetry (Yoyo mode)

e. Stays at the surface for recovery after a mission, providing frequent GPS fixes when requested (Recovery mode)

f. Sends multiple stored profiles to shore if previous attempts fail due to bad weather

g. Uses Kermit protocol (instead X-Modem) to transfer files

4. Control Hardware and Software Architecture

Hardware

Most of the EM-APEX control is performed by the APF9 microprocessor, with the exception of the electrically-isolated EM subsystem, which has its own microprocessor. The APF9 and accompanying memory (including RAM, Flash, and EEPROM) comprise a system with multiple fallback positions in anticipation of potential failure modes. Firmware is stored in Flash and loaded into RAM upon power up or wake from sleep. Current mission parameters are stored in RAM and EEPROM, in addition to the default parameter values built into the Flash firmware.

Serial ports

The APF9 board has six serial ports connected as follows:

- Console
- CTD
- Iridium
- COM1 – EM Subsystem
- COM2 – Spare
- GPS

Memory types available to APF9 (in order of increasing volatility)

- 512 K Flash memory (Flash denotes a faster type of EEPROM accessed in blocks) holds the APF9 firmware program, including the default parameter values. This Flash can only be changed by opening the float and using a special burner (JTAG) to load a new complete firmware.
- 2 K EEPROM (non-Flash) holds the current version of the mission parameters and can be updated manually (without opening the float) through the console `mission params` submenu. This is the version of the mission parameters that is used if the RAM copy becomes corrupted and needs to be reset.
- 8 MB Flash memory holds all processed and raw data files saved during the mission to be sent via Iridium. [***Warning***: *Care must be taken not to exceed the 8-MB capacity or the float will be unable to call in and the mission will fail.*] The current firmware does not have the capability to erase Flash blocks while a mission is in progress.

- 512 K battery backed up RAM holds most variables used by the operating code, including float state, EM processing steps, and the most current version of the mission parameters, including all updates since the mission started. This RAM is maintained even when the microprocessor is turned off (though not when the batteries are disconnected).
- 2 K RAM is built into the microprocessor and is used for temporary variables. These are not retained when the power is cycled.

Software

The APF9 program is organized into a series of mission states including descend, ascend, park, hibernate, telemetry, and recovery. Transitions among these states are directed by the `MissionControlAgent` subroutine in `src/control.c`, based on conditions and time intervals determined from a number of user-adjustable parameters (Appendix A). Commands issued to the EM-APEX through the console port (including testing and mission setup) are chosen from a series of menu options included in Appendix B. Once the command `mission start` has been given, the profiler enters autonomous operation, and further changes can be made through Iridium communications.

5. Opening Procedure (Pressure Case)

The pressure case will need to be opened to replace the batteries or reset the microprocessor manually. Reset is done by shorting pins (JP2) next to serial line on APF9 board.

The following steps should be done with the float supported in a horizontal position on a work bench. "Down," "up," "top," and "bottom," refer to the direction when the float is oriented vertically.

a. Undo the plug next to the antenna with a 3/16-inch hex driver

   [***Note***: *If an internal vacuum is present, air rushing in can be heard. Properly stored with an internal vacuum, air should not rush out; if this should occur, it may be a sign of released hydrogen gas and care should be taken. If there is any concern, the internal pressure can be read prior to opening the instrument. To read the internal vacuum, type* `c` *on the computer console.*]

b. Insert the long-handled 3/16-inch ball hex driver into the access port (Fig. 6) and loosen the internal tension rod

c. Disengage the tension rod and remove the upper end cap, taking care not to over-extend any internal cables

d. Disconnect the ribbon cables from the CTD and the coax cable from the antenna

e. Slide the bottom end cap (connected to the electronics chassis) downward and remove it from the pressure case, taking care not to overextend the cables connecting the end cap to the external electrodes

f. Reconnect the CTD ribbon cables and antenna coax cable to the top end cap. This will allow serial communication through the end cap clip leads and the operation of the CTD, GPS, and Iridium.

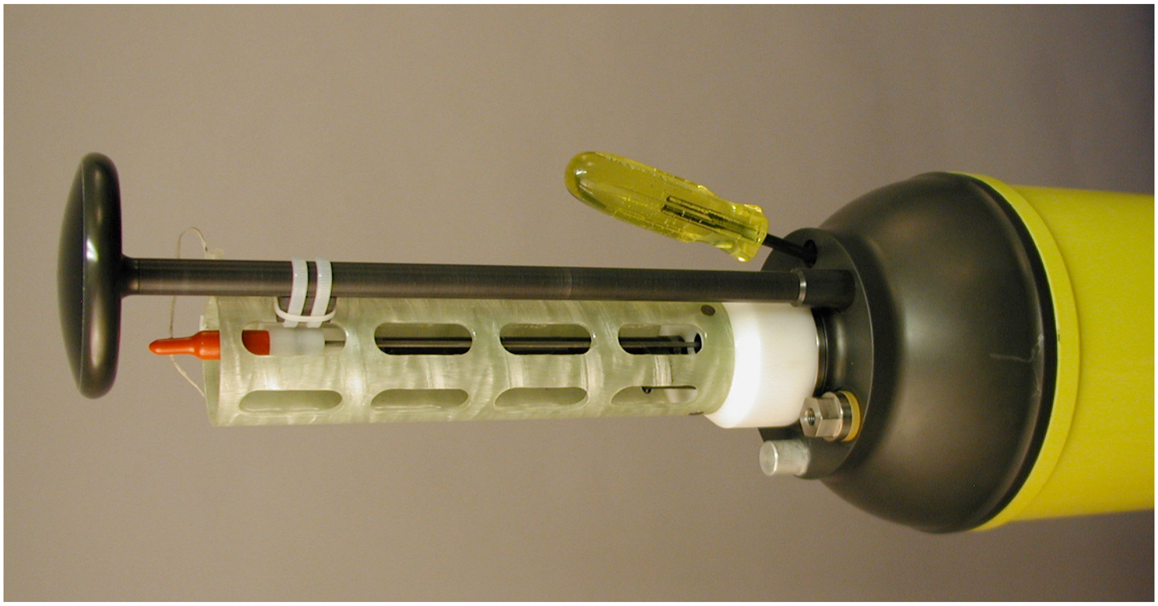

Fig. 6. Long-handled, 3/16-inch ball hex driver inserted into access port

6. Closing Procedure (Pressure Case)

Use a vacuum pump with an evacuation tool and place the 7/16-inch MS Plug on the short hex driver (Fig. 7).

Fig. 7. Vacuum gauge assembly

a. Slide the electronics chassis (with bottom endcap attached) into the pressure case tube until the bottom endcap seats onto the tube

b. Rotate the bottom endcap to the correct orientation (line up the marks on both the tube and end cap to ensure that the compass and electrodes are correctly aligned)

c. Reconnect the CTD ribbon cable and antenna coax to the top endcap

d. Slide on the top end cap, making sure that the threaded tension rod on the chassis fits into the threaded socket on the end cap

e. Use a long-handled 3/16-inch ball hex driver to pull the socket onto the threaded tension rod until the end cap and tube are flush. Do not tighten yet.

f. Rotate the top end cap to line up with the blue paint marks. Now tighten tension rod socket with hex driver.

g. Pull a vacuum of about –5" to –6" Hg using vacuum pump and vacuum gauge assembly (Fig. 7). Close off the pressure case by tightening the port plug with the hex driver while the vacuum pump is running.

h. Check the vacuum by pressing `c` on the computer console connected to the serial communication clip leads. Note that the internal air pump valve must be in "closed" position. If lower than –4.5" Hg, bleed air in gradually by loosening the port plug. Press `c` repeatedly to monitor the progress, then tighten the port plug again at –4.5" Hg. If higher than –4.5" Hg, return to step 'g' above to remove more air.

7. Maintenance

O-rings

O-rings are located between two mating surfaces in areas where water needs to be kept out. Small (half-inch) O-rings are located at the base of the antenna and on the top end cap access plug. Two others are about six inches in diameter and are located at the ends of the pressure tube.

All O-ring and mating surfaces should be inspected for cleanliness (especially human hair), nicks, slices, scratches, and dents.

Batteries

When the float will not be used for an extended period of time, the batteries should be disconnected from the APF9 and EM boards so as not to expend their energy by keeping the processors running. This must be done with the float disassembled by removing the plugs leading from the battery packs. The preferred order for unplugging is 1) APF9 power, 2) EM micro power, 3) EM preamp power (two cables, either order). Reconnecting the batteries should be done in reverse order. For long-term storage batteries should be removed to prevent acid leaks from damaging the instrument.

Electrodes

Little maintenance is needed on the electrodes themselves, but the agar plug connecting them to the external seawater should not be allowed to dry out while the float is in storage. When constructed, the agar is made with 35 psu saltwater, which can be prevented from evaporating by covering the agar with aluminum-coated Mylar tape. Note that the tape must be removed before deploying the EM-APEX because it is electrically insulating.

CTD

The primary concern with the CTD is not to allow saltwater to dry within the conductivity cell, as salt crystal formation tends to destroy the platinum electrode coating and change the conductivity measurement. Flushing with clean fresh water or de-ionized water is recommended after EM-APEX recovery. Once flushed, the CTD can be left either full of fresh water or dry. In either case, the rubber cap and plugs should be left on the CTD intake and exhaust ports to prevent contamination. However, these coverings must be removed before deployment.

8. Communications

Serial connection

Setup a PC terminal program to communicate at 4800 bits/s, 8 bits/character, with no parity, one stop bit, and no flow control. Some of the instructions below are specific to WinXP HyperTerminal. In the `File` menu select `Properties`. On `Settings` tab set the backspace key to emit `Ctrl+H` and the terminal emulation to `VT100`.

MS-DOS Kermit and C-Kermit, both from Columbia University, work well. The Kermit file transfer in HyperTerminal is not compatible so Xmodem must be used for file transfer with HyperTerminal.

Connect the PC serial line and EM-APEX current loop using the two small boxes (Fig. 8): a) Black Box (p/n CL090A-F) "RS-232 <--> current loop" converter and b) Integrity Instruments (p/n 232-OPT1), "optical isolator," which provides optical data transmission but no electrical connection between two RS-232 lines. The optical isolation prevents the EM-APEX pressure case and internal circuit common ground from being driven to +12 VDC as it would without the isolator.

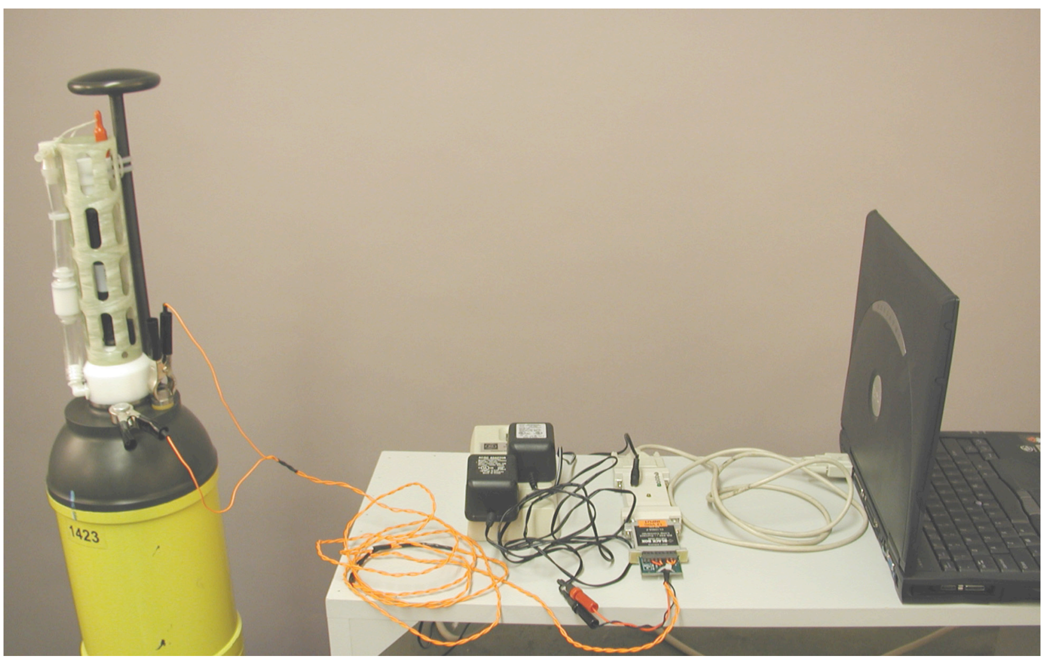

Fig. 8 Serial communication clip leads

Connect the alligator clips (refer to Fig. 8) to the EM-APEX as follows:

Attach the first current loop clip to the bare aluminum post on the upper end cap. Next, remove any white plastic protective anti-static guard from the adjacent stainless steel nut, and attach the other current loop clip to the nut. This nut is insulated from the end cap with a thick white plastic washer and is electrically connected (inside the float) to the case of the pressure transducer, which, in turn, is connected with a blue wire to the hot lead of the 20-mA current loop on the APF9 board.

> [***Warning***: *Connect to the insulated hex nut* ***after*** *connecting to the grounded aluminum post to prevent static discharge that might permanently damage the pressure transducer. Note that the anti-static guard must be removed before launching for proper ballasting. Make every effort not to touch the nut during launch to avoid a static discharge.*]

Within 2 s of attaching the second clip, the EM-APEX should emit a banner as follows:

```
EM-APEX FwVer=2a050615 Float=XXXX State=0 Dbg=2 Log=C,4800 Psim=0
HWsim=0

NewCmd>
```

At this point the commands from the main menu can be entered. For a list, type `?`.

Iridium

At-sea mission modification

Mission changes can be made anytime the EM-APEX calls home via the Iridium phone. Mission changes are done by uploading a file that modifies the mission parameter list held in the float's RAM. The file is uploaded at the beginning of each call to the server, so it is possible for a shore user to respond to engineering or environmental data as quickly as one cycle period later (i.e., the next time the float surfaces).

Data transmission

CTD and velocity profiles are returned via the Iridium satellite phone at each surfacing, following GPS position acquisition. The EM-APEX places a call to a land-line modem and executes a Kermit script on the server computer. The standard behavior is to transfer all files from the most recent profile working backwards through previous profiles that have not yet been sent, up to a preset maximum connection time. For small files, the resulting data transfer rate may depend as much on the number of files as on the total quantity of data. Thus the current system is more efficient for deep profiling.

9. Ballasting

The EM-APEX comes from the factory with its air weight adjusted and marked on the tube as computed from ballasting. The float needs to be neutrally buoyant; thus users must provide the manufacturer with the water properties and the depth of the operation area. Because the float will arrive already ballasted, users need only check that the weight on the delivered float is as specified by WRC. Users should weigh the float upon arrival, and then if any internal changes are made to the float, make certain there is no change in the total weight when the float is put into the water because the weight must remain the same as it was from the factory. This is only true if the changes made are all internal. Any external changes will change the volume of the float, which may require factory re-ballasting.

10. Use Procedures

Please refer to Appendix C for a sample checklist that encompasses many of the procedures described here.

The following steps are taken to setup and deploy an EM-APEX.

Arrival checkout

Unpack and weigh the EM-APEX to see if what was received is identical that marked on the tube at the factory (the balance must sensitive to 1–2 g out of a total of 28,000 g). Check that agar plugs covering electrodes are still intact and saturated with saltwater (i.e., have not shrunk back appreciably from the external PVC diameter). Replace Mylar tape or other covering over electrode ports. Check that plugs are covering CTD intake and exhaust and that the conductivity cell is either dry or filled with fresh water. Check that the anti-static cap is covering the stainless steel communications post.

Simulation mode test

Simulation mode can and should be used to test a wide variety of EM-APEX components and sample missions. The instructions below assume that the simulation is being run on land, but nearly the same procedure can be followed for testing at sea prior to deployment.

Procedure

a. Connnect serial line hardware together: RS-232 cable, opto-isolator with power supply, black box serial-to-current loop converter with power supply, and orange twisted wires with clips to attach to the float (see Fig. 8)

b. Run GPS for 20–30 minutes with *`show gps`* to allow GPS to determine its new location and satellite position. Disconnect the console wiring and carry the float outside where it has an unobstructed view of the sky and let it run for more than four hours. It should run the GPS receiver for 40 minutes to insure the almanac is current. Then send data via Iridium or transfer with the "data transfer" command. Examine the GPS information sent for position and the quality of the satellite reception; six to eight satellites are necessary for optimal performance.

c. Set clock with "old" command *`t`* (see Appendix B for more details)

d. Set parameters to `default` with certain modifications as indicated in the "simulation" column of Table 1, Appendix A, or as desired. See Appendix B for instructions on setting parameters. For testing, set the `FloatId` to an unused float number. You may want to use 9xxx where xxx is the last three digits of the real float number so that the simulation files are separate from the real data. With firmware 2a050615 there is a flaw preventing simulation runs greater than 200 dbar (decibar) because the simulated CTD profile only goes to 200 dbar. Actual profiles have been run to 500 dbars with this firmware. The maximum working depth of the first generation EM-APEX pressure hull is 1,000 dbar.

e. Start mission and leave with a view of the sky for 10 hours or more

f. Stop mission by typing *`<ctrl-C>terminate<enter>`* followed by *`mission stop`* and confirmed with *`Y`*

g. Confirm stop by putting float to `sleep` and waking up again (any key). EM-APEX splash line should come up with `State=0`

h. Check battery *`c`* and flash *`flash status`*

i. Transfer data: begin at 0, end at the latest Flash point location, 1 file, no mission updates, Xmodem, port C, type in timeout number (the value suggested should work)

j. Begin transfer on HyperTerminal. Filename format: `flash_FFFF_MMDD` (FFFF= float serial number; MMDD=month/day)

Sample console output

The following is the console output for a simulation.

After parameters are entered as described in Appendix B, use *`q`* then *`Y`* to write the mission to EEPROM:

```
nam=val> q

mission bytes changed = NNN out of 390

Write mission to EEPROM? [Y/N] Y

Copying mission to EEPROM
```

Force the APF9 to sleep. The subsequent wake up causes a complete reboot.

```
 NewCmd> sleep

APF9 power off for 21600 s
```

Wake up with any key:

```
EM-APEX FwVer=2a050615 Float=9888 State=0 Dbg=2 Log=C,4800 Psim=1
HWsim=0
```

Note that the `Psim=1` indicates that a simulation is running (the value of the `simulate_pressure` parameter is 1).

```
NewCmd> mission start

2005/07/17 14:22:14   2276 CmdMode()  Attempting to start mission

Attempting to start mission -- Please wait.

2005/07/17 14:22:15   2277 AirPumpRun()    t=1 s, 11.78 V, 234
mA, 0.17 "Hg

2005/07/17 14:22:26   2288 SelfTest() Vacuum  Passed:  internal
pressure [104, 0.2"Hg] less than threshold [106, 0.8"Hg].

2005/07/17 14:22:28   2290 Sbe41GetP  rtm=0.723  n=10  buf=[    -
0.04\000\n] sim=1 *p=0.00 ret=1

2005/07/17 14:22:28   2290 SelfTest()  SBE       41       passed,
pressure=0.00

2005/07/17 14:22:30   1 PTableInit()  pTableSize=220    Ptop=4.00
Pbot=984.00

2005/07/17 14:22:31   2 PreludeInit() Mission  started.  [PrfId=1,
itime=2290]

2005/07/17 14:22:31   2 PistonMoveAbsWTO()  077->031  076  075  074
073 072 071 070 069 068 067 066 065 064 063 062 061 060 059 058
057 056 055 054 053 052 051 050 049 048 047 046 045 044 043 042
041 040 039 038 037 036 035 034 033 032 031 [290 s, 11.32 V, 165
mA, CumT=290 s]

Mission started

2005/07/17 14:27:21   292 ProperPowerOff()  AlarmSec=300

2005/07/17 14:27:21   292 check_stack()    heap  now:  0,  never
used: 19266, max stack: 2250

APF9 power off for 7 s

2005/07/17 14:27:22   293 Apf9PowerOff()    for   7   s,   vote=1
use=293 dt10=0 dt12=0 dt13=0 dt23=0
```

```
2005/07/17 14:27:30   301 SimulatePressure()     state=1    t=301
P=31.3  T=20.930  S=36.628  sis=25.89  dpdt=0.163  dt=8.0  dnf=66.2
upf=0.0 ct=294 na=0

2005/07/17 14:27:31   302 Sbe41GetP   rtm=0.711  n=10  buf=[     -
0.05\000\n] sim=1 *p=31.46 ret=1

2005/07/17 14:27:32   303 PreludeRun() Prelude  is  done.   P=31.46
datenow=2005/07/17 14:27:31

2005/07/17 14:27:32   303 ProperPowerOff()  AlarmSec=311

2005/07/17 14:27:32   303 check_stack()     heap   now:   0,   never
used: 20698, max stack: 818

APF9 power off for 7 s

2005/07/17 14:27:33   304 Apf9PowerOff()    for    7    s,    vote=1
use=304 dt10=0 dt12=0 dt13=0 dt23=0

2005/07/17 14:27:41   312 SimulatePressure()     state=1    t=312
P=33.1  T=20.754  S=36.629  sis=25.94  dpdt=0.161  dt=8.0  dnf=65.1
upf=0.0 ct=298 na=0

2005/07/17 14:27:41   0 DescentProfInit()   Level2Flag=0
LowerP=180.0 DateNextLevel2=0, was=0

2005/07/17 14:27:42   1 DescentProfInit()   YoyoFlag=0  UpperP=0.0
LowerP=180.0

2005/07/17 14:27:42   1 DescentProfInit()   RawEfSaveFlag=1

2005/07/17 14:27:42   1 DescentProfInit()   Profile=1    initiated
at mission-time 312 s.

2005/07/17 14:27:43   2 Sbe41GetP     rtm=0.715  n=10  buf=[     -
0.04\000\n] sim=1 *p=33.39 ret=1

2005/07/17 14:27:44   3 DescentProfInit()   First pressure: 33.39

2005/07/17 14:27:48   7 PistonMoveAbsWTO()  031->052  032  033  034
035 036 037 038 039 040 041 042 043 044 045 046 047 048 049 050
051 052 [127 s, 11.24 V, 177 mA, CumT=127 s]
```

Diagnostics to check while and after running a simulation

a. Kermit log files on your local server (see Section 11 below) will say when floats have called in and run scripts

b. Check files uploaded to the home server (`gps`, `gpa`, `ctd`, `efp`, `efr`)

c. Dump the Flash file in the `~emapex/data-in directory` after the mission ends (includes all of the above files). See Sections 11 and 12 below.

Real mission set-up

The setup for a real mission is the same procedure as the Simulation Mode Test above, except that users should use their real float number xxxx, and be sure that `simulate_pressure=0` and `simulate_hardware=0`.

Set values for the rest of the mission parameters as desired using the `mission parameters` menu as described in Appendix B. Sample real mission parameter values can be found in Appendix A, Table 2, "EDDIES Mission." Check the real time clock and the internal vacuum (Appendix B). Before starting the new mission, be sure that simulation mode is off:

```
NewCmd> sleep
```

Wake up with any key and make very certain `Psim=0` and `HWsim=0`

```
EM-APEX FwVer=2a050615 Float=9888 State=0 Dbg=2 Log=C,4800 Psim=0
HWsim=0
```

Then start with *`mission start`*. The console output should look like the simulated mission except the simulated p, T, and S values are not used.

Watch the console screen until piston is returned to the ballast point. The float is now ready to launch (see below).

Final or pre-launch checkout of physical float

a. Cover all the holes in the bottom of the cowling to prevent silt from entering the cowling and changing the ballasting

b. Remove tape from the five electrodes so that the agar surface is exposed

c. Remove the rubber plugs and cap from the CTD intake and exhaust

d. Remove the plastic static guard from the serial communications post

Launch

Launching can be done using a draw hitch (also known as the highwayman's hitch) to attach the lifting line to the float on the upper pressure case (see Fig. 9). This knot can be untied by tugging on the release end once tension is removed. When bringing the knot under tension care must be taken that the knot does not invert and pull the release bight through. Once under tension, the knot should hold well and then release when the float is in the water and tension is removed. The performance of the draw hitch can vary with different rope thicknesses and pliability, so be sure to test with adequate backup before using for an actual launch.

Lower the EM-APEX into the water and release the draw hitch with a slip line (see Fig. 10). After the release of trapped air the float should sink within several tens of seconds. In several hours EM-APEX will resurface and phone home with the first data.

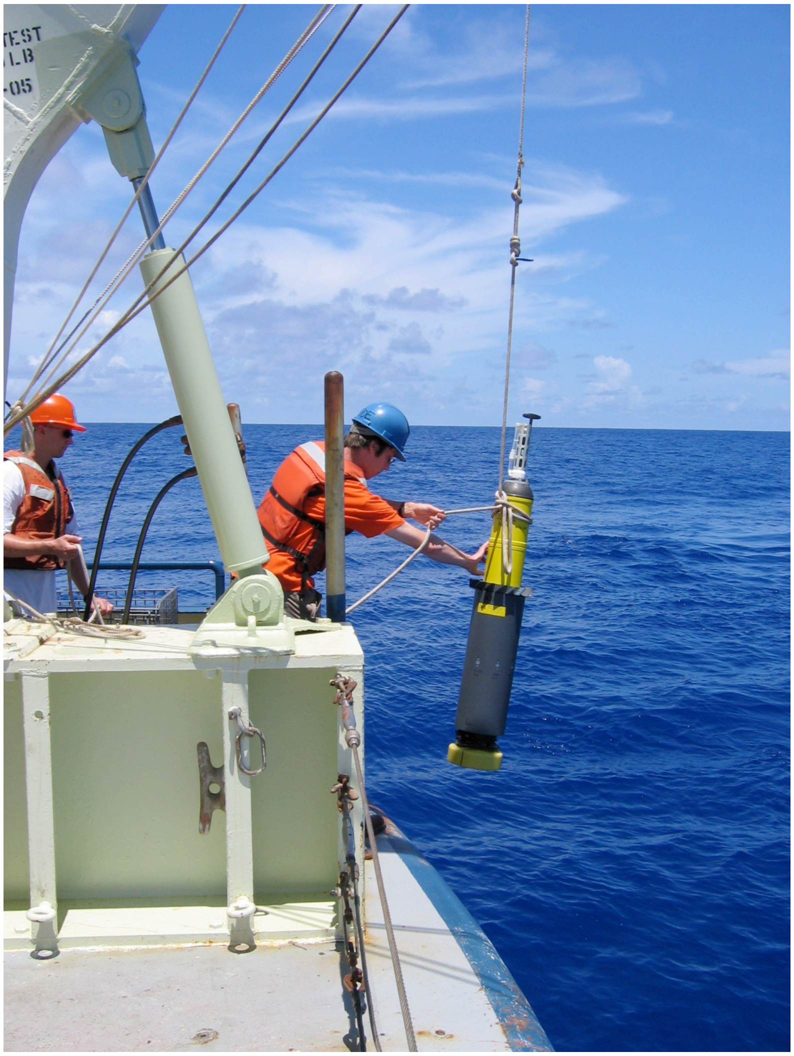

Fig. 9. Launching the EM-APEX with a draw hitch

Fig. 10. EM-APEX after launch and prior to sinking

Real time mission changes

After deployment, EM-APEX mission changes can be made by modifying a file on the server computer called `~emapex/MISSION_UPDATES_####` (where #### is the four-digit float serial number). This file contains mission parameter/value pairs to be updated in the float's RAM and is uploaded to the float automatically at the start of each login. Changes become effective immediately, and can even influence the quantity of data sent and time spent on the surface for the current connection. The file is created by the `chmis` utility, which adds a checksum to each line of the ASCII input file `source_file` to create a second ASCII file `change_file`, which must then be renamed to

`MISSION_UPDATES_####` with the proper float serial number and placed in the `~emapex` home directory on the server. `source_file` contains a list of mission parameter/value pairs of the form: `parameter_name(parameter_value)`.

Float recovery

EM-APEX recovery from the water can be accomplished by hand from a small boat or with a noose as shown in Fig. 11. The wire noose is attached to a flexible recovery line, which is fed through a lifting point such as a block or an A-frame. The noose and the line are attached to a pole with masking tape allowing the pole to tear free when the line is pulled taut. The 60-lb weight of the EM-APEX makes lifting and carrying by one person possible, though a second set of hands is often valuable.

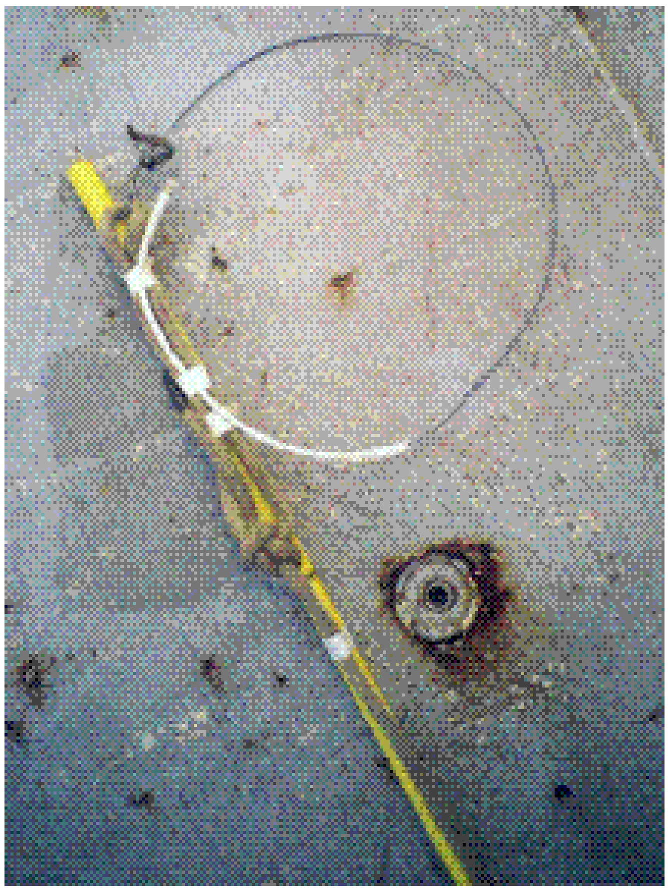

Fig. 11. Recovery noose

Mission halt

[***Important***: *Follow the commands in the order as indicated below.*]

To stop the mission, re-connect the console interface (see Fig. 8). When the float is in a profiling mode, it sleeps about four seconds every 15 seconds. It also sleeps for five minutes at a time in the hold mode. Press any key while the float is sleeping to cause it to wake up into the console mode.

If the EM-APEX is running, the sequence *<Ctrl-C>terminate<enter>* will cause the float to sleep. Press any key to wake up into the console mode.

Once in the console mode, stop the mission using the command `mission stop` and respond with *y* to confirm. The screen should display:

```
NewCmd> mission stop

Warning: Stopping the mission renders the float non deployable until the mission is started again.

Stop the mission anyway? Type y to stop the mission

WARNING: Mission stopped.

The EM-APEX is not deployable until the mission is restarted.
```

Extracting data from Flash

Connect to the 20-mA current loop on the console port from the leads on the computer to the EM-APEX (see Fig. 8). Ensure that the mission is stopped as described above. Use the command `data transfer`. Do not include the mission updates file. Use X-Modem 1-K with HyperTerminal. If using Kermit set the PC to receive 4000-byte packets with block check type 3. The data transfer rate is 200–300 bytes/s. The following is an example with C-Kermit on a Linux system.

```
NewCmd> data transfer

test_ConnectAndSend -- flashWriteIndex=2312814

Enter flashWriteIndex_beg: 0

Enter flashWriteIndex_end: 2312814

Enter number of files for fwi beg to end (1-10) : 1

Include mission_updates_file (y/n): n
```

```
Enter Xmodem or Kermit (x/k): k

Enter Kermit block check type (1/2/3): 3

Enter port (C=console, 1=serial1, 2=serial2, I=Iridium): c

Enter filename prefix: jun18a

Enter seconds allowed for entire transfer [11564]: 20000

Really Start now? (s=start, b=bail): s

Configure and start Kermit receive.  Type:

<Alt-X>    for command mode on Kermit-95

<Ctrl-\> C for command mode on C-Kermit

set send pause 10

set send packet 1000

set receive pause 10

set receive packet 4000

receive

^A0 S~H @-#Y3~*!*j"^A%9Y)1X^O
```

11. Home Server

During the course of an EM-APEX mission, data files are sent via Iridium phone calls to a modem connected to the user's server computer. (Note that in this manual, the machine "henry" has been used for demonstration purposes.) Requirements for setting up a server include: a) one or more dedicated phone lines and modems; b) the capability to run automated scripts (e.g., via the Unix `cron` system); and c) Kermit file transfer software installed.

With each phone call, the float logs into the server as user "emapex," downloads a new `MISSION _UPDATES` file (if available) and uploads any unsent files in the data Flash (up to a pre-defined connection time limit). These file transfers are made via Kermit using server scripts in the `~emapex/bin` directory (and called by the APF9 firmware subroutines in `phonehom.c`).

On henry, a 1-GHz Pentium III machine, running RedHat Linux, two automated scripts are used: 1) `rtgps.run` is run by `cron` every minute to check for new uploaded data files and send an e-mail (or text message) with the most recent surfacing position and time to the specified account(s); 2) `emarun.m` (running in a Matlab shell) runs `pseudoflash.m` every hour to compile all data files into a replicate of the float's Flash memory and decodes these (with `ema2mat.c`), then creates velocity and CTD profile plots (with `emavel.m`) for viewing via a web browser.

To set up a server, follow the procedures in Appendix D.

12. Data Format

Data is saved in the 8-MB Flash memory as a set of consecutive files of different types. The filename convention once these are transferred to the server is:

`ema-FFFF-TTT-PPPP-YYYYOODD-HHMMSS_BBBBBBB_EEEEEEE`

where `FFFF` is the EM-APEX serial number (e.g., `1632`); `TTT` is one of five file types (`ctd`, `efp`, `efr`, `gps`, `gpa`); `PPPP` is the profile number, with odd numbers corresponding to down profiles and even numbers corresponding to up profiles; `YYYY` is the year, `OO` is the month, `DD` is the day, `HH` is the hour, `MM` is the minute and `SS` is the second that the file was written (usually at the end of the profile); `BBBBBBB` and `EEEEEEE` are the beginning and ending Flash addresses for the file (in bytes).

`ctd` files contain CTD data from the SBE-41, `efp` files contain processed electric field data from the EM subsystem, `efr` are large files containing raw (1 Hz) electric field data (including accelerometer and magnetometer output), `gps` files contain positions and information on fix quality from the start of each surfacing interval, `gpa` files contain GPS positions from the end of each surfacing interval (after the Iridium call), for use in estimating surface drift.

For example, the file `ema-9636-ctd-0005-20050721-130505_0548374_0549468` contains CTD data from the fifth profile ("down" portion of the third round trip) by float #9636 (a simulation name for float 1636), was written at 13:05:05 UTC on July 21, 2005, and occupies 1094 bytes of memory starting 548,374 bytes into the Flash (i.e., at the time of writing the file, approximately 1/16 of the 8 MB storage capacity had been used).

Data within each file is in binary format. Details can be found within the APF9 source code (in `efpro.c` and `emapro*.c`) or in the `ema2mat.c` decoder program. Variable names in the `.mat` output match the C source code names whenever possible.

13. Data Decoding and Processing

The C program `ema2mat.c` accepts complete individual files as they are sent over Iridium or will accept the concatenation of many of those files. However, if the phone connection is dropped in the middle of a transfer, some files will be broken in several places. The matlab script `pseudoflash.m` assembles all the phoned home files from a single directory into files that look like the APF9's Flash, and then runs `ema2mat.c` on the files.

The Matlab script `emavel.m` combines data from the `ctd` and `efp` files to compute velocity profiles and make plots. The script `emarun.m` runs `pseudoflash.m` then `emavel.m` repetitively forever, thus providing a web page that is periodically updated with new profiles.

14. Appendix A: Mission Parameters

Built into each EM-APEX are three lists of mission parameters: Default, Office 1, and Office 2. Table 1 lists all 105 mission parameters, together with a sample value and a brief description. Times are in seconds. Piston positions are in counts. Air pressure/vacuum values are also given in counts, and water pressures are reported in dbar. A description of how the parameters relate to the float mission behavior is presented in Fig. 12. See Table 2 for other sample mission configurations (sets of parameter values).

Table 1. Parameter names, default values, and descriptions

FloatId	1424	integer serial number of float (use 9xxx for simulations)
MaxAirBladder	120	air pressure counts never to exceed
OkVacuumCount	96	Use 106 for simulation without pressure casePistonBuoyancyNudge 8 for incremental speed change
PistonDeepProfilePosition	56	for going deeper after park
PistonFullExtension	215	counts never to exceed
PistonFullRetraction	31	counts never to be less than
PistonInitialBuoyancyNudge	15	for initial motion from neutral buoyancy
PistonParkPosition	51	for parking
PistonStoragePosition	31	same as ballast position
PnpCycleLength	255	prevents usage of Pnp
PressurePark	500	pressure desired when parked
PressureDeep	600	after park, go deeper before ascending
TimePrelude	0	keep this at 0 until he firmware is tested
TimeDescentProf	16,050	timeout for DescentProf including max TimeHold*
TimeDescentPark	12,500	timeout for DescentPark
TimeDescentDeep	2500	timeout for going to PressureDeep after Park
TimeDown	24,259	timeout for DescentPark
TimeUp	4650	add to TimeDescentProf to finish telemetry
TmoAscent	3750	added to TimeDescentProf to scram to sfc
simulate_pressure	0	construct a synthetic pressure record for lab testing (Note: MUST BE ZERO in ocean !!!)
simulate_hardware	0	must also be zero in ocean !!!
PreludeRepPeriod	100	time between pressure checks during prelude
PreludePressureThreshold	20	pressure when prelude ends
IdFirstLevel2	2	profile number for first Level2 profile (usually deeper)

TimeYoyoOnceBeg	0	time after prelude to start single subsurface yoyo
TimeYoyoOnceEnd	0	time after prelude to stop single yoyo
DateCycleStarted		updated by firmware
TimeCycleRep	43,200	period to increment DateCycleStarted
TimeCycleYoyoBeg	80,000	start yoyoing after DateCycleStarted
TimeCycleYoyoEnd	90,000	stop yoyoing
TimeCycleHoldLongBeg	172,800	start long holding after DateCycleStarted
TimeCycleHoldLongEnd	341,888	stop long holding
TimeHoldShort	6,150	duration of short holds
TimeHoldLong	169,088	duration of long holds
TimeLevel2Rep	0	seconds between repetitive Level2 profiles
PrTopYoyo	50	yoyo min pressure
PrBotYoyo	100	yoyo max pressure
PrBotLevel1	150	standard max pressure
PrBotLevel2	200	deeper profile every TimeLevel2Rep
RawEfSaveRep	2	save raw EF data every Nth profile
RawEfSendFlagProfiling	1	send raw EF data when profiling
RawEfSendFlagRecovery	1	send raw EF data during recovery
Vmin	0.05	minimum speed before piston is nudged
PistonInitAscentLevel2	90	counts for starting up from level 2
PistonInitAscentLevel1	93	counts for starting up from level 1
PistonInitDescentProf	68	counts for starting down from surface
PistonInitDescentProfFirst	67	first time counts for starting down from surface
DurationFastProfiling	0	time after prelude ends to switch to slow (park mode) profiling. Zero means never.
DateForRecovery	2005/10/01 00:00:00	pickup date desired
TimeForRecovery	0	as above but timed from prelude ending
RecoveryRepPeriod	900	wake up interval
RecoveryNRepConnect	1	phone home every Nth RecoveryRepPeriod
RecoveryIRepConnect	0	initial value for NRep counter
TmoConnect	300	modem connect and logon timeout
TmoXfrFastProfiling	300	data transfer timeout for fast profiling
TmoXfrProfiling	300	data transfer timeout for standard (park mode) profiles
TmoXfrRecovery	300	data transfer timeout for recovery
TmoXfrMin	300	minimum value allowed for transfer timeout
DatePreludeEnded		set by firmware
DateNextLevel2		set by firmware
ModemType	I	L for landline (office testing), I for Iridium
ModemBaudRate	19,200	use 19200 for Iridium
PhoneNumbers	0012062213080 0012062213248 0012065439844 0012065439845	calls each of these in order
PacketLength	4,000	bytes in Kermit packets sent home

TmoTelemetry	600	overall timeout for GPS & phoning home
debuglevel	2	2 is fine
logport_key	2	use C for console
logport_baud	115,200	only use 4,800 for console
EmaProcessNvals	50	time interval of raw EF in demod fits
EmaProcessNslide	25	time to slide demod fits
TmoGpsUpdate	2,400	longer GPS time-on every TimeGpsUpdateRep
TmoGpsShort	600	standard GPS on time.
TmoGpsAfter	300	GPS time after phone call
TimeGpsUpdateRep	432,000	the interval that GPS is on for a long time, about every 5 days
TimeGpsGrabRep	60	GPS sampling interval for saved positions
RawSvBytesStop	4,000,000	stop saving raw EF before 8MB FLASH fills
PressureAnticipate	7	expected overshoot when reversing direction
PressureNearSurface	7	ascending surface detection threshold
SigmaThetaFollow	26.27	the target sigma-theta potential density
PressureFollowDefault	75	used when unable to locate SigmaThetaFollow
PressureFollowMin	50	restricts the pressure range to follow
PressureFollowMax	100	restricts the pressure range to follow
PistonFollowDefault	81	if unable to locate SigmaThetaFollow in profile
SalinityBallastPoint	34.878	PSU, ballast point used
TemperatureBallastPoint	5.102	degC, with PistonStoragePosition
PressureBallastPoint	1000	dbar, to compute PistonFollow
PistonCountsPerCC	0.8650790	counts/cc piston scale factor
FloatMass	27,900	Grams, air weight (dry)
FloatAlpha	–2.72e–06	per dbar compressibility of hull
FloatBeta	7e–05	per degC temperature coefficient of hull
use_iPiston	0	MUST BE ZERO! code not finished
HeartBeatProf	5	usual time from power off to power on
HeartBeatHolding	300	wake up interval during hold mode
TimeOutNudge	8	max piston run-time during profile
CtdSamplePrFirst	7	first pressure in CTD grid
CtdSampleDelPr1	2	pressure interval in first CTD grid segment
CtdSampleDelPr2	5	second grid segment
CtdSampleDelPr3	10	
CtdSampleDelPr4	20	
CtdSampleN1	25	number of intervals in first CTD grid segment
CtdSampleN2	20	second segment
CtdSampleN3	35	
CtdSampleN4	25	
SimulationDpdtCoef	10	vertical velocity used for simulation

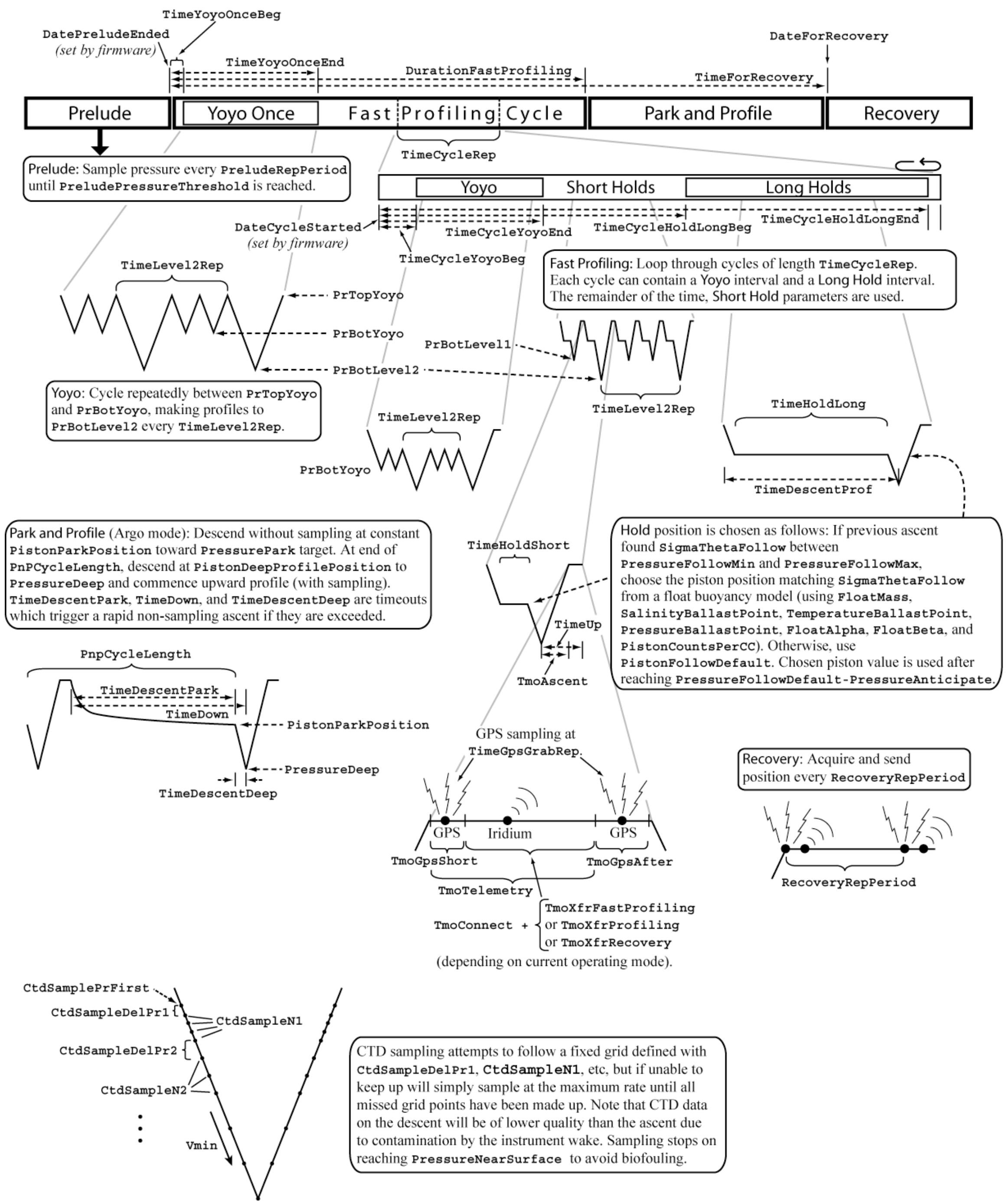

Fig. 12. Schematic view of the various EM-APEX mission stages and the role of the mission parameters in setting their duration and sequence. Individual diagrams represent graphs of pressure (i.e., depth) vs. time within various stages.

Table 2. Sample parameter values (– indicates the "Default" value)

Parameter	Default	Office 1	Office 2	Simulation	EDDIES Mission
FloatId	0000	9876	9872	9636	1636
MaxAirBladder	120	–	–	–	–
OkVacuumCount	96	106	106	–	–
PistonBuoyancyNudge	10	8	8	–	–
PistonDeepProfilePosition	56	–	–	–	–
PistonFullExtension	215	115	–	–	–
PistonFullRetraction	21	31	31	31	31
PistonInitialBuoyancyNudge	22	15	15	–	–
PistonParkPosition	51	45	57	–	–
PistonStoragePosition	31	–	–	–	–
PnpCycleLength	100	100	100	–	–
PressurePark	500	500	500	500	500
PressureDeep	600	600	600	600	600
TimePrelude	0	0	0	0	0
TimeDescentProf	10000	–	–	20853	20853
TimeDescentPark	12500	12500	12500	12500	12500
TimeDescentDeep	2500	2500	2500	2500	2500
TimeDown	43200	24259	24259	24259	24259
TimeUp	26400	–	–	10267	10267
TmoAscent	12000	–	–	16667	16667
simulate_pressure	0	1	1	1	–
simulate_hardware	0	–	–	–	–
PreludeRepPeriod	100	–	–	–	–
PreludePressureThreshold	20	–	–	–	20
IdFirstLevel2	0	–	–	2	2
TimeYoyoOnceBeg	0	–	–	–	–
TimeYoyoOnceEnd	0	–	–	–	–
DateCycleStarted	1970/01/01 00:00:00	–	–	–	–
TimeCycleRep	345600	43200	43200	–	–
TimeCycleYoyoBeg	0	10800	10800	–	–
TimeCycleYoyoEnd	0	21600	21600	–	–
TimeCycleHoldLongBeg	172800	28800	28800	–	–
TimeCycleHoldLongEnd	345600	43200	43200	341888	341888
TimeHoldShort	7200	100	1800	4186	4186
TimeHoldLong	172800	3600	3600	169088	169088
TimeLevel2Rep	0	–	–	–	–
PrTopYoyo	50	–	–	–	–
PrBotYoyo	100	–	–	–	–
PrBotLevel1	150	–	–	–	200
PrBotLevel2	200	–	–	170	500
RawEfSaveRep	10	2	5	–	–
RawEfSendFlagProfiling	1	–	–	–	–
RawEfSendFlagRecovery	1	–	–	–	–

Parameter	Default	Office 1	Office 2	Simulation	EDDIES Mission
Vmin	0.08	–	–	0.06	0.06
PistonInitAscentLevel2	77	90	102	–	–
PistonInitAscentLevel1	95	85	97	–	–
PistonInitDescentProf	47	45	57	–	–
PistonInitDescentProfFirst	46	40	52	–	–
DurationFastProfiling	0	–	–	–	–
DateForRecovery	2005/10/01 00:00:00	–	–	–	–
TimeForRecovery	0	172800	172800	–	–
RecoveryRepPeriod	1800	–	–	–	–
RecoveryNRepConnect	1	–	–	–	–
RecoveryIRepConnect	0	–	–	–	–
TmoConnect	300	–	–	–	–
TmoXfrFastProfiling	300	–	–	1800	1800
TmoXfrProfiling	900	300	300	–	–
TmoXfrRecovery	300	–	–	–	–
TmoXfrMin	300	–	–	–	–
DatePreludeEnded	1970/01/01 00:00:00	–	–	–	–
DateNextLevel2	1970/01/01 00:00:00	–	–	–	–
ModemType	I	L	–	–	–
ModemBaudRate	19200	4800	–	–	–
PhoneNumbers	0012062213080 0012062213248 0012065439844 0012065439845	13080,13248, 39844,39845			
PacketLength	4000	–	–	–	–
TmoTelemetry	600	–	–	1800	1800
debuglevel	2	–	–	–	–
logport_key	C	2	2	–	–
logport_baud	4800	115200	115200	–	–
EmaProcessNvals	50	–	–	–	–
EmaProcessNslide	25	–	–	–	–
TmoGpsUpdate	1800	600	600	2400	2400
TmoGpsShort	620	300	300	1800	1800
TmoGpsAfter	200	120	120	–	–
TimeGpsUpdateRep	1209600	36000	36000	691200	691200
TimeGpsGrabRep	60	30	30	30	30
RawSvBytesStop	6000000	–	–	4000000	4000000
PressureAnticipate	7	–	–	–	–
PressureNearSurface	7	–	–	–	–
SigmaThetaFollow	24	26.27	26.27	26.25	26.25
PressureFollowDefault	75	–	–	85	85
PressureFollowMin	50	–	–	–	–

Parameter	Default	Office 1	Office 2	Simulation	EDDIES Mission
PressureFollowMax	100	–	–	120	120
PistonFollowDefault	62	74	77	75	75
SalinityBallastPoint	34.878	–	–	–	–
TemperatureBallastPoint	5.102	–	–	–	–
PressureBallastPoint	1000	–	–	–	–
PistonCountsPerCC	0.8650790	–	–	–	–
FloatMass	27900	–	–	27957	27957
FloatAlpha	-2.72e-06	–	–	–	–
FloatBeta	7e-05	–	–	–	–
use_iPiston	0	–	–	–	–
HeartBeatProf	5	–	–	–	–
HeartBeatHolding	300	–	–	–	–
TimeOutNudge	8	–	–	–	–
CtdSamplePrFirst	4	–	–	–	–
CtdSampleDelPr1	1	–	–	–	–
CtdSampleDelPr2	5	–	–	–	–
CtdSampleDelPr3	10	–	–	–	–
CtdSampleDelPr4	20	–	–	–	–
CtdSampleN1	150	–	–	–	–
CtdSampleN2	20	–	–	–	–
CtdSampleN3	25	–	–	–	–
CtdSampleN4	25	–	–	–	–
SimulationDpdtCoef	0.02	–	–	–	–

[***Note:*** *Default, Office 1, and Office 2 are lists of parameter values obtained by typing* `default`, `office 1`, *or* `office 2` *in the* `mission params` *menu. Simulation values were used in experimental testing. EDDIES values are the final mission parameters used for a float in the EDDIES (Eddies, Dynamics, Mixing, Export, and Species composition) experiment.*]

15. Appendix B: Terminal Operations

Terminal operations involve a variety of commands chosen from a series of menus. The commands `help` or `?` list the commands in the current menu. Most of the menu item names are self-explanatory.

Main menu items

```
air pump – runs air pump for 6 s

air valve close,  air valve open

cal (or c) -- shows battery status, air pressure, piston position

ctd p,  ctd stp,   ctd version

data transfer

debug level

ema show,  ema wakeup,  ema sleep

flash status,  flash zero,  flash format

gps show

mission params,  mission start,  mission stop

modem check

piston go,  piston stop

simulation params

sleep -- lowest APF9 power

?? -- more tests
```

Check float status

The most common menu command is `cal` or `c` to check the battery voltage and current, internal air pressure, and piston position.

```
NewCmd> c

Bat: 12.69 V,  34.7 mA, AirP: -0.08 "Hg, Piston: 58 c
```

The voltage for new batteries is greater than 14 V while the internal air pressure should be more negative than –4.4" Hg for proper operation of the air bladder.

Real time clock

The following shows how to set and check the APF9 "real time clock"—the time base used for all float operations and saved data. These commands do not appear in the main menu but are part of a submenu accessed by typing *old*. Type *n* to return to the main menu when finished. Try to set the clock to within one or two seconds of UTC (coordinated universal time). Press <*enter*> about 3 s before the time entered to allow for 3 s of processing time in the float.

```
NewCmd> old

OldCmd> t 2005/06/14 14:30:10

Real time clock: "2005/06/14 14:30:10"

OldCmd> t

Real time clock: "2005/06/14 14:30:17"

OldCmd> n

NewCmd>
```

Set mission parameters

Parameters are set using the `mission params` submenu. The following example shows the contents of the menu as well as some sample parameter changes. Note that parameter changes entered here are not saved into the EEPROM until the *quit* command has been issued and acknowledged with *y*.

```
NewCmd> mission params

name=value -- entry for mission parameters

list       -- show mission parameters

default    -- use default mission parameters

office 1   -- use office 1 simulation mission

office 2   -- use office 2 simulation mission

quit (q)   -- quit this menu
```

`nam=val>` *`office 2`* (sets mission parameters for office test 2)

```
setting mission parameters for office test 2
nam=val> FloatId=9888
confirm  FloatId=9888
nam=val> PnpCycleLength=255
confirm  PnpCycleLength=255
nam=val> TimeDescentProf=16050
confirm  TimeDescentProf=16050
nam=val> TimeUp=4650
confirm  TimeUp=4650
nam=val> TmoAscent=3750
confirm  TmoAscent=3750
nam=val> TimeCycleRep=345600
confirm  TimeCycleRep=345600
nam=val> list
FloatId                  = 9888
MaxAirBladder            = 120
OkVacuumCount            = 106
PistonBuoyancyNudge      = 8
PistonDeepProfilePosition = 56
PistonFullExtension      = 215
PistonFullRetraction     = 31
...
```

[***Note***: *For a complete list of all mission parameters see Appendix A*.]

16. Appendix C: EM-APEX Checkout

1. Measure Air Weight (As received) [g] ____________
2. Enlarge mounting holes in ema3.9 (mainboard) ____________
3. Verify Electrode wiring colors ____________
4. Verify LNEF2A input wiring colors ____________
5. Connect LNEF2A input wire harness to feedthru pigtail ____________
6. Connect LNEF2A power wires ____________
7. Connect LNEF2A Output Coax Cables ____________
8. Label E1 and E2 on ema3.9 (mainboard) ____________
9. Label Electrode wire color and EF Channel on PVC skin ____________
10. Connect ema3.9 - to - APF-9I serial cable ____________
11. Connect ema_orn IDC ribbon cables to ema3.9 connectors ____________
12. Wire connectors to AA battery packs ____________
13. Verify proper wiring of Iridium phone -to- APF-9I ____________
14. Lock-down GPS wiring with RTV ____________
15. Lock-down Iridium phone wiring with RTV ____________
16. Verify Iridium SIM card is in place and locked-down ____________
17. Disconnect displacement pump motor (H-drive connector) ____________
18. Measure Initial Piston Position as delivered [mm] ____________
19. Download APF-9I firmware (Mission Software) ____________
20. Install default Mission in EEPROM ____________
21. Write Float ID into EEPROM ____________
22. Force Shutdown from Console ____________
23. Wake-up APF-9 and note voltage, current draw, air press. ____________
24. Verify RAM version of Mission software matches EEPROM ____________
25. H-drive disconnected + new firmware verify piston position ____________
26. Reconnect H-drive motor wires and verify proper operation ____________
27. Note voltage and current to move piston out one count ____________
28. Note voltage and current to move piston in one count ____________
29. Format FLASH (20 min.) {No bad blocks?} ____________

30. Test GPS for serial cable connectivity
31. Power Iridium Phone and listen for RF relay click
32. Send AT command to Phone (Responds with OK)
33. Test Air Pump (listen for operation)
34. Close Air Solenoid
35. Run air pump for 3 each 6-s intervals & verify bladder pressure
36. Open air solenoid (should hiss)
37. Set realtime clock in APF-9 & verify operation
38. Verify comm. Link to ema3.9 board
39. Verify operation of SBE41 CTD
40. Connect all battery packs to circuit boards
41. Verify low current draw to all circuits in "hibernate" mode
42. Dress All internal wiring with cable ties and RTV (or Hot Melt)
43. Measure static H fields with gaussmeter
44. Attach APF-9I -to- SBE41 CTD ribbon cable
45. Attach Iridium phone RF coax to antenna feedthru
46. Close-up pressure case
47. Pull partial vacuum in pressure case with special WRC tool

Calibrations

Calibrate Tilt Sensors

Calibrate Compass (outside)

Verify GPS operation (outside)

Verify Iridium phone operation (outside)

Subassembly Inventory

Piston Assy. S/N

Piston Motor S/N

APF-9I, Iridium Phone, GPS Assy. S/N

APF-9I S/N

ema3.9 S/N

SBE41 CTD S/N

SIM #

Phone #

17. Appendix D: Server Setup

Server and processing software are available from WRC as four gzipped tar files: `emahome.tgz`, `emadec.tgz`, `emaauto.tgz`, and `chmis.tgz`. This configuration requires a machine running RedHat or Fedora Linux using C and Matlab. Source code is included to aid in porting to other operating systems.

Steps to follow:

Unpack the files with the command *`tar xzf <filename>`* after *`cd`*'ing into the correct directory as described below.

Use `mgetty` and `sendfax` for emapex logins from several land-line modems. The setup on henry includes 8 serial ports (allowing control of up to 8 modems) using a Comtrol RocketPort Octable .

Make a new user `emapex` with home directory `/home/emapex` using a bash shell. Set the password the same as the password in the embedded software, `emaproc/src/phonehom.c`. See line 808, the call to `login()`.

Login as `emapex`. This is the directory for logins from the floats. Unpack `emahome.tgz` in `/home/emapex/`. Use the command *`make init`* to initialize a few files. The files in `/home/emapex/bin` are called by the embedded software.

The rest of the software is installed in your home directory, not `/home/emapex`.

Login as yourself.

Unpack `emadec.tgz` in `~/emapex/emadec/`. The command *`make install`* compiles and links `emadec.c` and copies the executable to your `~/bin` directory, which should be included in your `$PATH` .

Unpack `chmis.tgz` in `~/emapex/chmis/` . This is used to change the mission in the floats. Use the command *`make test`* after modifying the `source_file`. The resulting file, `change_file`, must be renamed and put into the `/home/emapex` directory with names `MISSION_UPDATES_NNNN` where `NNNN` is the float ID number.

Unpack `emauto.tgz` in `~/emapex/emauto/`. Use the command *`make init`* to make some directories used by the Matlab scripts. Run Matlab in this directory with the command *`emauto`*. It processes and plots the data once an hour. Point your web server at the `png` directories to allow others to see the graphs.