

Description of one dive cycle for the S2-X floats

```
// descend to desired depth
get_vacuum(&AD_VAC);
open_oil_valve();
open_air_valve();
while ( wait for Fall_Tm seconds ) {
    sleep(100);           // wake up every 100 seconds to check pressure
    check_pressure();
    if ( our_first_pump && pressure > dBarGo ) { // only do this once
        pump_oil(Tlast); // pump OUT the amount we best-guessed from before
        store_pump(Below100);
    }
    if (pressure > z_target) { // pump if we are too deep
        pump_oil(t); // where t is calculated as a function of the float's
        store_pump(SINKING); // vertical velocity
    }
}

get_vacuum(&AD_VAC);
check_pressure();
close_air_valve();
close_oil_valve();

// multi_seek - try to get closer to target depth
for ( Nseek ) { // # of seeks to try
    check_pressure();
    if ( too_deep ) {
        pump_oil(t); // make correction for t < mxSeek seconds
        store_pump(SEEK); // uses the dTadZ parameter to compute t
    } else { // too shallow
        open_oil_valve(t); // uses the dTsdZ parameter to compute t
    }
    sleep(STLmin); // settle out for SETTLE mins
}

// drift - wait deep & avg P & T
for( Nsam ) { // take Nsam samples
    scan_p_t_s(&p,&t,&s);
    // seek each time we record a sample
    {
        if ( p > (z_target + 100) ) { // float is too deep
            pump_oil(t); // make correction for t < mxSeek seconds
            store_pump(SEEK_DURING_DRIFT);
        } // NOTE: Float does not correct if it is too shallow
    }
    sleep(SAMmn); // every SAMmin minutes get a recording of P, T, S
}
check_pressure();

// fall is just like descend but with different parameters
```

```

// ascend - rise to the surface
    check_pressure();
    start_collecting_samples();
    pump_oil(PmpBtm);
    store_pump(START_OF_RISE);
    while ( wait for Rise Time to elapse or until stop rising ) {
        sleep(100);    // wake every 100 seconds
        check_pressure();
    }
    stop_collecting_samples(); // Stop CTD & store bin averaged profile data

    check_pressure(); // surface pressure before filling the air bladder
    pump_air();

// surface - xmit data, check air bladder & pressure offset
    check_pressure();
    pump_oil(MxSfP);    // pump oil, unless sum of the dive's pumps exceed MxHiP
    store_pump(END_OF_RISE);
    for ( time < IRIssec ) {
        transmit_data();
        sleep();        // wait until time for next transmit
    }
    check_pressure();    // check pressure to see if need to reset offset
    if ( SBE pressure needs to be corrected for drift ) {
        pressure_reset();    // sleep for ~3 mins as this happens
        check_pressure();
    }

```