Prototype V1 (Satellite Tag)

Sensors:

* Wet/Dry (also initial turn on)
* Temperature
* Pressure (Depth)
* PAR
* Tilt
* ARGOS communications

Phase 0:

* Does nothing much other than establish what the depth is.
* This lasts 24 hours to allow the tag to be positioned at the bottom without accidentally triggering the sampling process.

Phase 1:

* Waits for the tag to start ascending.
* Logs depth/temperature/PAR/angle from vertical every 30 seconds (creates 1 message every 2 minutes).  It maintains two running medians of depth readings: a highly-smoothed measure based on the last few minutes, and a fast measure based on the last few seconds.  The ascent phase is triggered when the fast measure is more than 1m shallower than the slow one.  The idea is to allow for some jiggling about but still be able to react to the genuine ascent within a few seconds.

Phase 2:

* The ascent
* Logs depth/temperature/PAR/angle from vertical at 4Hz and creates 1 message per second.  Stops when the fast depth measure is no longer 1m shallower than the slow one (should be within a couple of minutes of reaching the surface).

Phase 3:

* The surface.
* Transmits every 40 seconds for about 40 days.  The wet-dry value at the time of transmission is inserted into the transmission, but is not used to control the transmissions.  Logging is as for phase 1.

Prototype V2 (Arduino with RockBlock Mk2 Iridium Module)

Sensors:

* Temperature
* Pressure (Depth)
* PAR?
* Accelerometer (Tilt Angle)
* GPS
* Iridium Communications

Phase 1:

* Logs until specific date/time for Burn Wire Release
* 1 record every 6 hours for up to 170 Days
  + Temperature
  + Pressure (Depth)
  + Tilt Angle
  + PAR
  + RTC Time
  + Battery Voltage
* If time is within 6 hours of burn wire time, set alarm to burn wire time. Otherwise increment alarm by 6 hours

Phase 2:

* Waits for Ascent to surface
* Wakes up at predetermined time to be synchronized with burn wire release (~30 min before? – burn wire error is +/- 2 mins per month = 12 mins, release takes 10 to 20 minutes to complete erosion)
* Establishes baseline depth
* Stays on continuously
* Records sample every (60 seconds?)
  + Temperature
  + Pressure (Depth)
  + Tilt Angle
  + PAR
  + RTC Time
* Waits for depth to change by (3m?)

Phase 3:

* Ascent to Surface
* Stays on continuously for (5 mins?)
* Records sample every (0.25 seconds?)
  + Temperature
  + Pressure (Depth)
  + Tilt Angle
  + PAR
  + RTC Time
* Once Complete, Set Sample interval to + 15 min

Phase 4:

* Logging Under Ice/Waiting to Arrive at Surface
* Records sample every (15 min?)
  + Temperature
  + Pressure (Depth)
  + *Max tilt angle over 30 second period*
  + PAR
  + RTC Time
  + Battery
* Power up GPS, wait (max 60 seconds?). See if GPS data is available. If no, go back to sleep. If Yes, proceed to Stage 5.

Phase 5:

* At Surface/transmission mode
* Records sample every hour
  + Temperature
  + Pressure (Depth)
  + *Max tilt angle over 30 second period*
  + PAR
  + RTC Time
  + Battery
  + GPS Latitude
  + GPS Longitude
  + Time to Acquire GPS Fix (max 60 seconds)
  + GPS Hdop
  + GPS Timestamp
  + Iridium Strength of Signal?
* Power up GPS, wait (max 60 seconds?). See if GPS data is available. If no, go back to sleep. If Yes, transmit Iridium message
  + Send one timestamp only
  + Check for messages in queue, if proper message has been sent, transmit all data (until next stop message is received). If no message waiting, go back to sleep.

Design considerations:

* Need Mosfet switches for GPS and Iridium to limit power during underwater phases
* Log all data to HEX file and store marker for how much data has been transmitted
* 9/12/16 to 3/1/16 is 170 days (max duration for release)
* Release only holds 40 lbs…
* Need large buffer for data messages – probably need Arduino mega