# Carbon Waveglider

MBARI PIC Firmware



During a ***measurement cycle***, several ***conditions*** are measured:

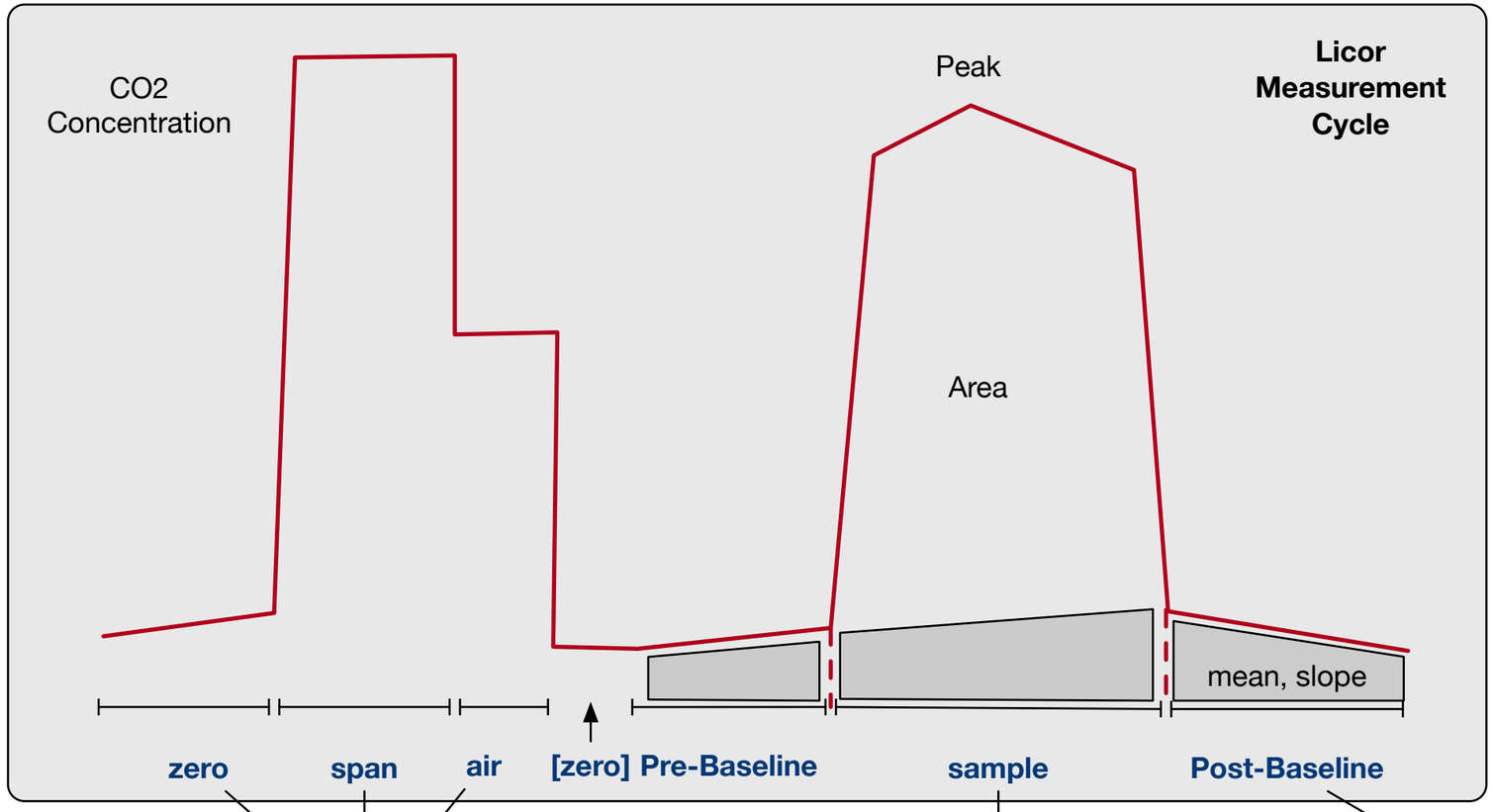
**zero**
**air**
**span**
**baseline (pre/post sample)**
**sample**

For each condition, multiple Licor samples are taken, containing several ***parameters***:

**co2, co2raw, co2ref, celltemp, cellpres, h2o, etc.**

Various ***measurements*** are taken for a set of parameters, which varies per-condition:

**Mean, Sdev, area, slope, etc.**




**zero, air, span**

sample N=10

| ***Measurents*** | ***Parameters*** |
|---|---|
| Mean, Sdev | co2, co2raw, co2ref, celltemp, cellpres, h2o |

**sample**

sample N=60

| ***Measurents*** | ***Parameters*** |
|---|---|
| Mean, Sdev, Peak, Area | co2 |
| Mean | co2raw, co2ref, celltemp, cellpres, h2o |

**baseline**

sample N=5

| ***Measurents*** | ***Parameters*** |
|---|---|
| Mean, Sdev, slope, min, max, endpoints? | co2 |

# Carbon Waveglider

MBARI PIC Firmware

During a ***calibration cycle***, several ***conditions*** are measured:

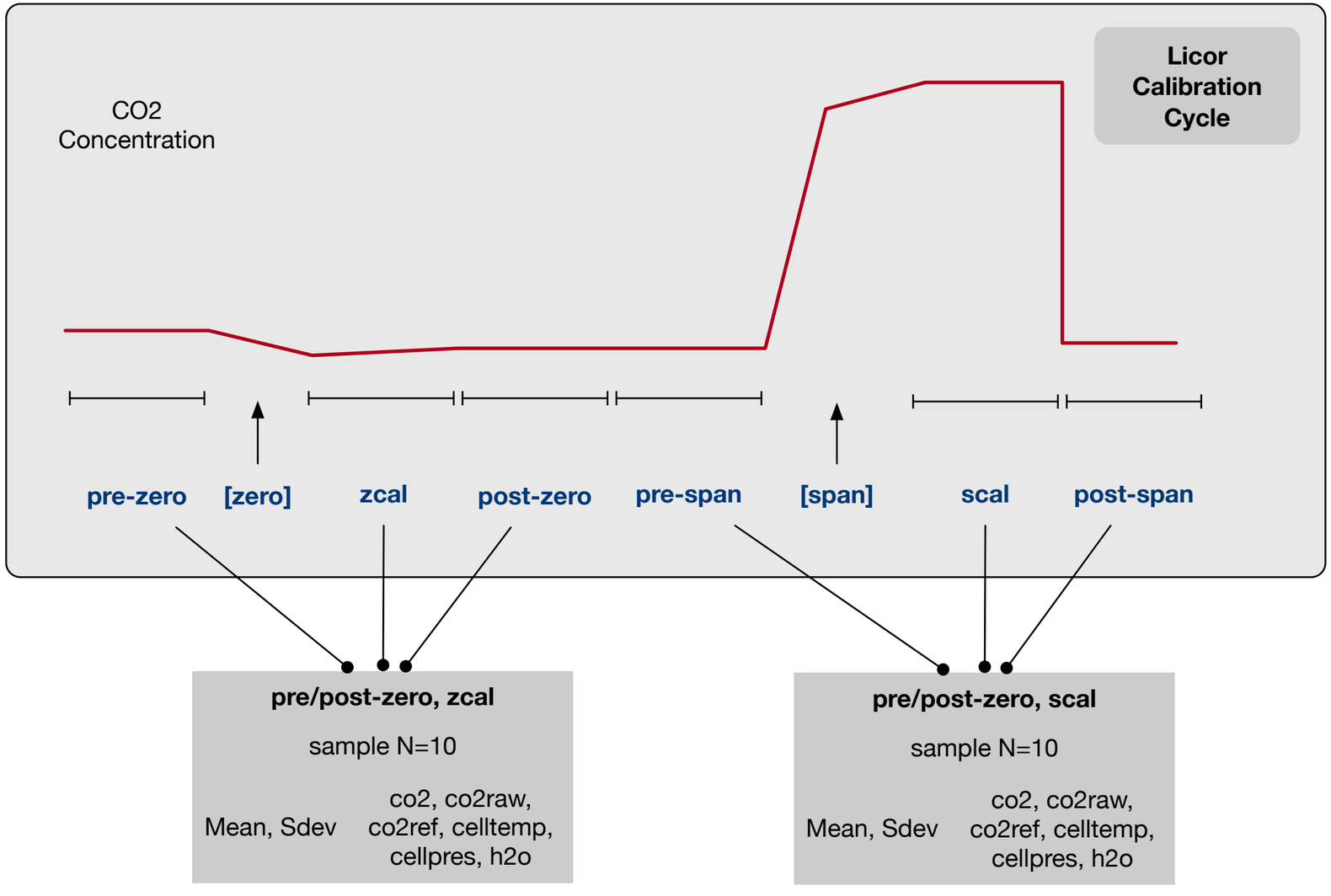
**pre/post-zero**
**zcal**
**pre/post-span**
**scal**

For each condition, multiple Licor samples are taken, containing several ***parameters***:

**co2, co2raw, co2ref, celltemp, cellpres, h2o, etc.**

Various ***measurements*** are taken for a set of parameters, which varies per-condition:

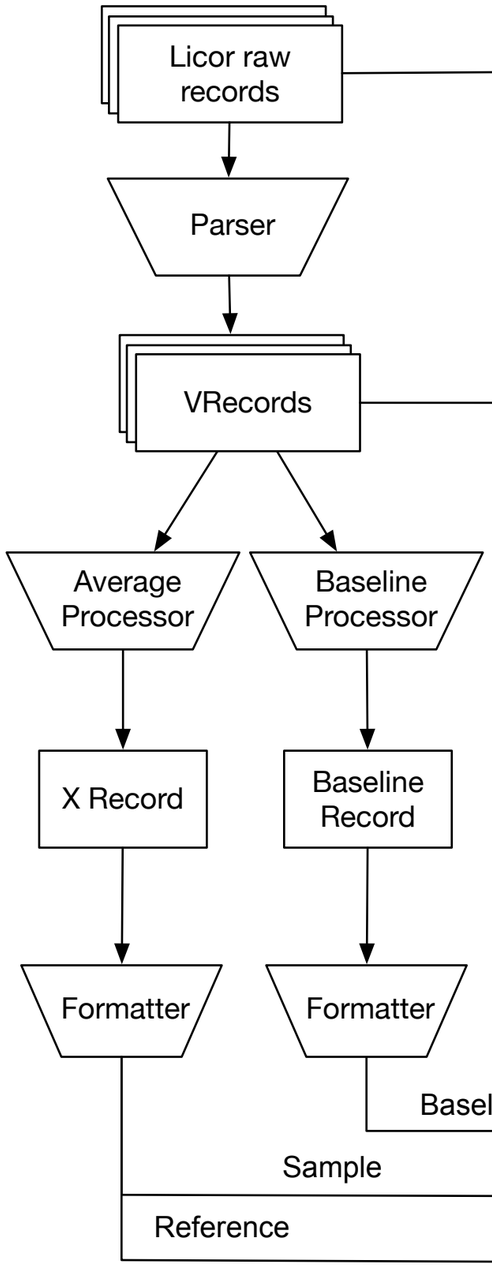
**Mean, Sdev, etc.**



# Carbon Waveglider

MBARI PIC Firmware

Data Flow




```
<li850><rs232><co2>true</co2><co2abs>true</co2abs><h2o>true</h2o><h2oabs>true</h2oabs><h2odewpoint>true</
h2odewpoint><celltemp>true</celltemp><cellpres>true</cellpres><ivolt>true</ivolt><flowrate>true</
flowrate><strip>false</strip><echo>false</echo><flowrate>true</flowrate><raw>true</raw></rs232></li850>
```

**Value Record**
(licor_vrecord_t)

| |
|---|
| timestamp |
| type |
| ctx |
| seq_n |
| licor fields[…] |

**XRecord**
(licor_xrecord_t)

| |
|---|
| timestamps[2] |
| type |
| ctx |
| seq_n |
| sample_N |
| Mean (vrecord) |
| Sdev (vrecord) |

**Baseline Record**
(licor_brecord_t)

| |
|---|
| timestamps[2] |
| type |
| ctx |
| seq_n |
| sample_N |
| oflags |
| endpoints[2] |
| mean value |
| sdev value |
| min value |
| max value |

**Raw records** are parsed into data structures (**value records**)

Value records contain timestamped raw data

Multiple value records stream through a **processor** that operatate on the record stream to make **measurements**

**Measurement records** (e.g. xrecord, brecord) contain measurement values and context (state) variables.

Measurement record content is formatted into various CSV **record types** for logging and display

Baseline

```
2019-05-17T12:13:27Z,B,M,0,10,1558095207500,-6.978840e+00,1558095212000,-6.911270e+00,-6.979,-6.910,-6.945,0.024
```

Sample

```
2019-05-17T12:13:24Z,S,M,0,3FFFFF,30,-7.136703e+00,1.215683e-01,3.029106e-03,5.228151e-05,1.209025e+01,4.071662e-03,…
```

Reference

```
2019-05-17T12:12:53Z,R,S,0,3FFFFF,8,-9.109272e+00,7.732845e-01,3.879632e-03,3.334937e-04,1.197710e+01,3.320222e-02,…
```

# Carbon Waveglider

MBARI PIC Firmware



## DATyymm.LOG

licor measurement, calibration samples

| timestamp | type | fields… | CRLF |
|---|---|---|---|

```
2019-05-02T08:49:12Z,R,Z,0,3FFFFF,8,8.453497e-01,1.305093e-02,-3.495046e-04,5.394967e-06,1.120606e+01,4.896278e-04,8.841268e+00,5.731668e-04,8.717278e-02,2.007264e-06,2.105246e+01,2.839897e-03,1.1
2019-05-02T08:49:18Z,R,S,0,3FFFFF,8,8.891398e-01,2.744835e-02,-3.675563e-04,1.133208e-05,1.121109e+01,6.558900e-04,8.847525e+00,7.058359e-04,8.719429e-02,2.460473e-06,2.107103e+01,2.764259e-03,1.1
2019-05-02T08:49:23Z,R,A,0,3FFFFF,5,9.592422e-01,9.107444e-03,-3.964901e-04,3.751769e-06,1.121623e+01,8.642814e-04,8.854718e+00,1.093234e-03,8.722152e-02,4.237782e-06,2.108585e+01,1.160358e-03,1.1
2019-05-02T08:49:26Z,B,M,0,10,1556786966500,9.310020e-01,1556786971000,9.717710e-01,0.930,0.989,0.968,0.023
2019-05-02T08:49:48Z,S,M,0,3FFFFF,30,9.540361e-01,2.030799e-02,-3.942457e-04,8.379639e-06,1.123179e+01,4.155738e-03,8.874358e+00,5.291691e-03,8.729175e-02,1.939195e-05,2.113332e+01,1.130981e-02,11
2019-05-02T08:49:51Z,B,M,0,10,1556786991500,9.246020e-01,1556786996000,8.760600e-01,0.871,0.945,0.911,0.027
```

## RAWyymm.LOG

raw licor samples

| timestamp | licor_xml | CRLF |
|---|---|---|

```
2019-05-02T08:46:21Z,<li850><data><celltemp>2.04872e1</celltemp><cellpres>1.01731e2</
cellpres><co2>-4.00649e1</co2><co2abs>1.7684102e-2</co2abs><h2o>7.50644</
h2o><h2oabs>6.5430402e-2</h2oabs><h2od
ewpoint>3.04880</h2odewpoint><ivolt>1.2276172e1</ivolt><raw><co2>4553910</
co2><co2ref>5034416</co2ref><h2o>2313719</h2o><h2oref>3949373</h2oref></raw><flowrate>0</
flowrate></data></li850>

2019-05-02T08:46:21Z,<li850><data><celltemp>2.04899e1</celltemp><cellpres>1.01734e2</
cellpres><co2>-3.33666e1</co2><co2abs>1.4568984e-2</co2abs><h2o>7.80770</
h2o><h2oabs>6.7351999e-2</h2oabs><h2od
ewpoint>3.60613</h2odewpoint><ivolt>1.2273654e1</ivolt><raw><co2>4453537</
co2><co2ref>4910523</co2ref><h2o>2251940</h2o><h2oref>3849542</h2oref></raw><flowrate>0</
flowrate></data></li850>

2019-05-02T08:46:22Z,<li850><data><celltemp>2.04926e1</celltemp><cellpres>1.01732e2</
cellpres><co2>-2.78307e1</co2><co2abs>1.2044548e-2</co2abs><h2o>8.07316</
h2o><h2oabs>6.9018244e-2</h2oabs><h2od
ewpoint>4.08099</h2odewpoint><ivolt>1.2276172e1</ivolt><raw><co2>4409710</
co2><co2ref>4854821</co2ref><h2o>2212811</h2o><h2oref>3790063</h2oref></raw><flowrate>0</
flowrate></data></li850>

2019-05-02T08:46:22Z,<li850><data><celltemp>2.04940e1</celltemp><cellpres>1.01738e2</
cellpres><co2>-2.33802e1</co2><co2abs>1.0048329e-2</co2abs><h2o>8.30910</
h2o><h2oabs>7.0486187e-2</h2oabs><h2od
ewpoint>4.49261</h2odewpoint><ivolt>1.2276172e1</ivolt><raw><co2>4392959</
co2><co2ref>4834863</co2ref><h2o>2188667</h2o><h2oref>3755033</h2oref></raw><flowrate>0</
flowrate></data></li850>
```

## SYSyymm.LOG

status information

| timestamp | msg | CRLF |
|---|---|---|

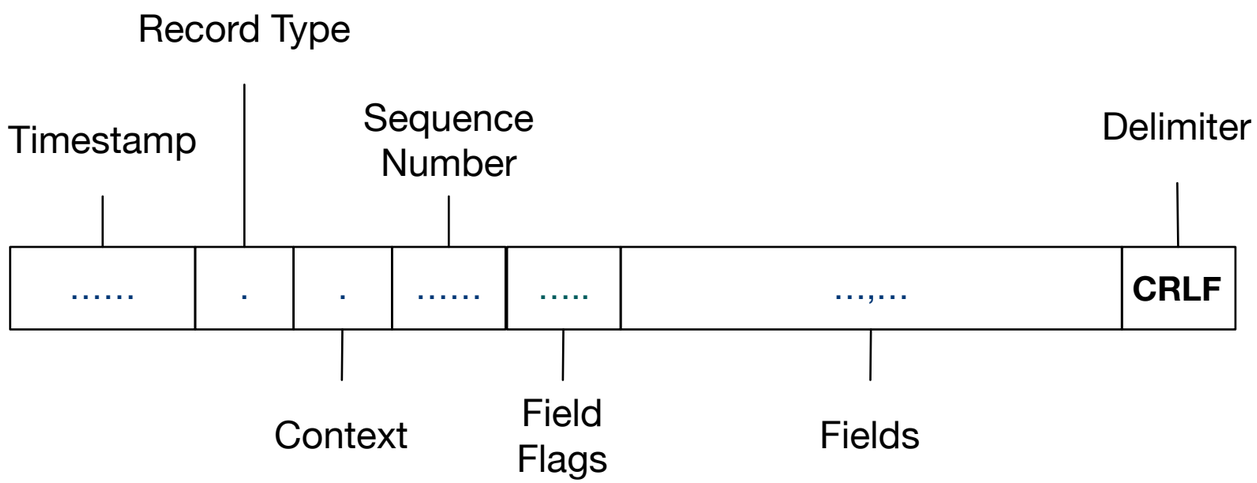
```
2019-05-20T08:37:57Z,nmeas.a streamx err[1]
2019-05-20T08:38:04Z,nmeas.a streamx err[1]
2019-05-20T08:38:09Z,nmeas.a streamx err[1]
2019-05-20T08:38:36Z,nmeas.a streamx err[1]
2019-05-20T08:38:44Z,set [BITS P* V*]
2019-05-20T08:38:45Z,set [BITS P* V* LI]
2019-05-20T08:42:43Z,set [BITS LI]
2019-05-20T08:42:49Z,or set err[1]
2019-05-20T08:42:50Z,set [BITS LI]
```

## Summary

| | |
|---|---|
| File Encoding | ASCII, UTF-8 |
| Record Format | CSV |
| Record Delimiter | CRLF (0x0D0A) |
| Timestamp Formats | ISO8601, Posix |
| File Names | DOS (case insensitive, 8.3) |
| Segmentation | monthly |
| Indexed | No |
| Record IDs | Self-identifying |

# Carbon Waveglider

MBARI PIC Firmware




| | | |
|---|---|---|
| **Timestamp** | ISO8601 and/or Epoch MS | `2019-05-20T08:58:41Z` |
| **Record Type** | Byte indicating type of record | `B: baseline R: ref S: sample V:value ?:default` |
| **Context** | Byte indicating record context | `Z: zero S: span A: air M: meas ?:default` |
| **Sequence Number** | Sample number | `32-bit unsigned integer` |
| **Field Flags** | Bit field identifying included fields | `0x0 - 0xFFFFFFFF` |
| **Fields** | comma-separated data | `format varies` |
| **Delimiter** | record separator | `CRLF (0x0D0A)` |

Example:
baseline record

```
2019-05-02T08:49:51Z,B,M,123,10,1556786991500,9.246020e-01,1556786996000,8.760600e-01,0.871,0.945,0.911,0.027
```

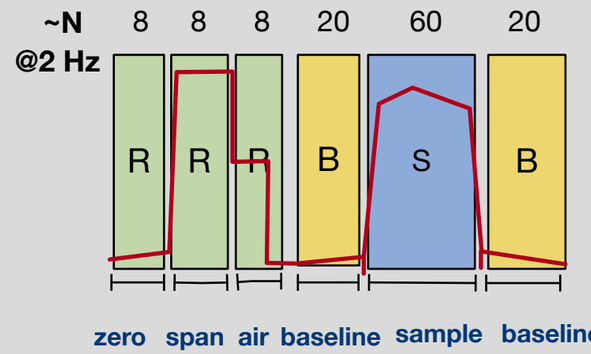
# Carbon Waveglider

MBARI PIC Firmware

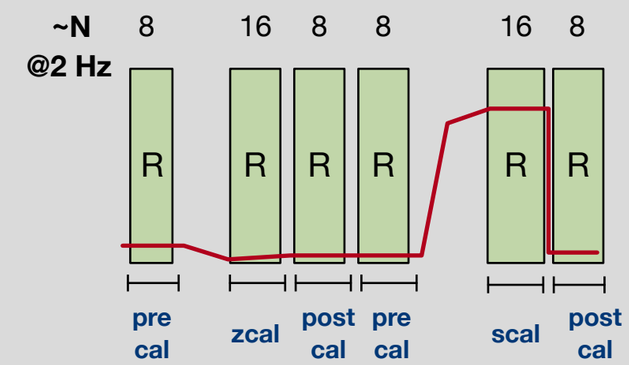
Record Type Content

| | | Field | Content |
|---|---|---|---|
| **Value Record** | V | **0:5** | **VALUE** [ co2, co2raw, co2ref, celltemp, cellpres, h2o ] |
| **Reference Record** | R | **0**<br>**1:12** | **N** sample_N<br>**MEAN, SDEV** [ co2, co2raw, co2ref, celltemp, cellpres, h2o ] |
| **Baseline Record** | B | **0**<br>**1:5** | **N** sample_N<br>**MIN, MAX, MEAN, SDEV, SLOPE** [ co2 ] |
| **Sample Record** | S | **0**<br>**1:6**<br>**7:9** | **N** sample_N<br>**MEAN** [ co2, co2raw, co2ref, celltemp, cellpres, h2o ]<br>**SDEV, PEAK, AREA** [ co2 ] |

**Measurement Cycle**



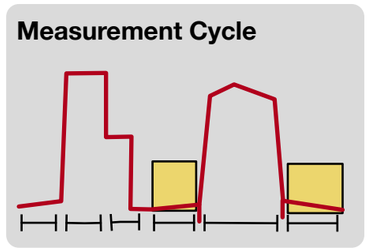

**Cal Cycle**

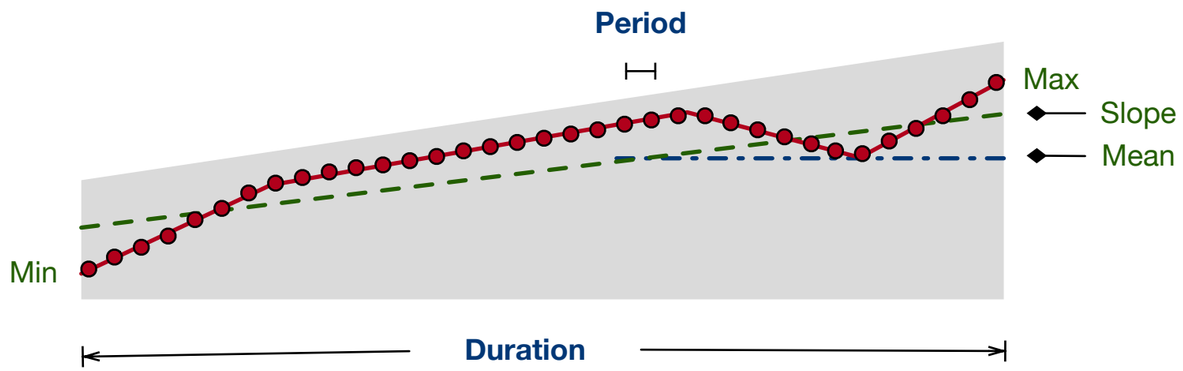


# Carbon Waveglider

MBARI PIC Firmware






## Baseline Measurement

### Inputs

Period : >=0.5
Duration : N / t / F()
Dest. : H|C|L

### Outputs

slope : dCO2/dt
Min : CO2
Max : CO2
Mean : CO2 [1]
Sdev : CO2 [1]

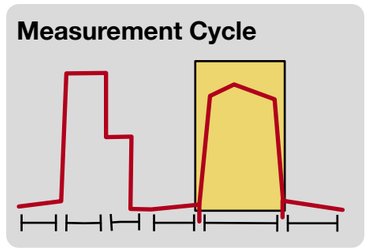
[1] - average computed using Welford's method (single pass, i.e. uses only current, previous samples, reduces over/under flow)

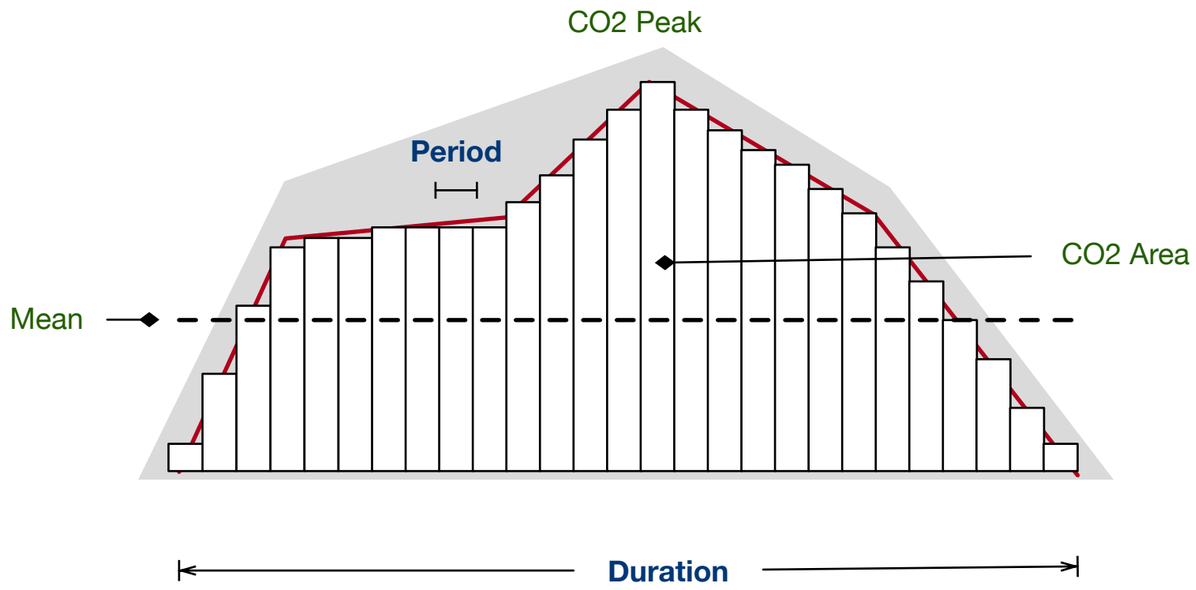
Baseline CO2 averages are taken at the beginning and end of measurement cycles to characterize sensor bias drift.

# Carbon Waveglider

MBARI PIC Firmware






**Average Measurement**

**Inputs**

Period : >=0.5 s (<= 2 Hz)
Duration : N / t / F()
Dest. : H|C|L

**Outputs**

Mean : all [2, 3]
Area : CO2 [1]
Peak : CO2

Averages of all parameters are taken for all conditions.

During the sample measurement, CO2 peak and area features are also measured.

[1] - Trapezoid method; dA = dy*dt = 0.5(y+yo)*(t-to)

[2] - measured parameters
co2, co2abs, h2o, h2oabs, h2odp, celltemp, cellpres, ivolt, flowrate, raw_co2, raw_co2ref, raw_h2o, raw_h2o_ref
DON't need h2oabs, h2o_raw, h2oref, h2odewpoint
DO need co2raw, co2ref, celltemp, cellpres, co2, h2o
MAYBE need co2abs?

[3] - average computed using Welford's method (single pass, i.e. uses only current, previous samples, reduces over/under flow)

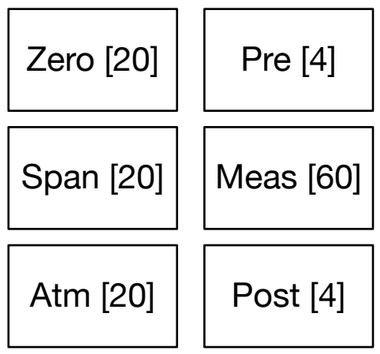
# Carbon Waveglider

MBARI PIC Firmware

Data Volume

## Raw Records

**46 kB/cycle**



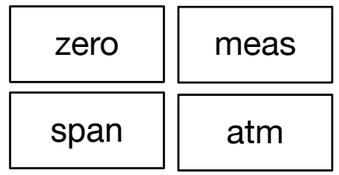

| | size | total |
|---|---|---|
| **raw** | ~365 bytes | **46720 bytes** |

Raw records occupy SD Card

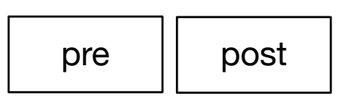
## Parsed Records

**0.8 kB/cycle**

x_record [4]




bline_record [2]



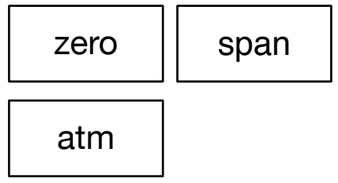

| | size | total |
|---|---|---|
| **bline_record** | ~55 bytes | **110 bytes** |
| **x_record** | ~180 bytes | **720 bytes** |

Parsed records occupy RAM

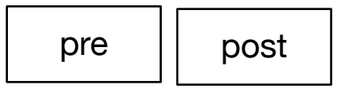
## CSV Log Records

**1.2 kB/cycle**

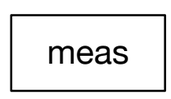
Type R [3]



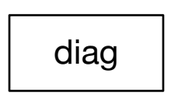

Type B [2]




Type S [1]




Type D [1]




| | size | total |
|---|---|---|
| **B** | ~115 bytes | **230 bytes** |
| **R** | ~180 bytes | **540 bytes** |
| **S** | ~410 bytes | **410 bytes** |
| **Diag** | ~65 bytes | **65 bytes** |

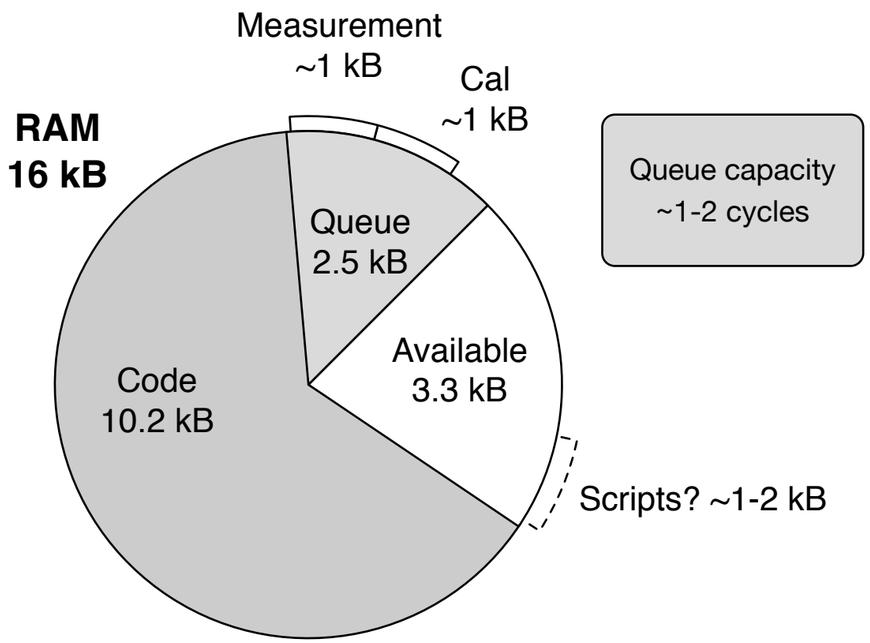
Log records occupy SD Card

# Carbon Waveglider

MBARI PIC Firmware

| | **Intermittently Powered** | **Continuously Powered** |
|---|---|---|
| **Sampling** | scheduled, immediate, synchronous | unscheduled, buffered, asynchronous |
| **Cycle logic** | Host implements | Instrument implements |
| **Sample Logging** | optional | required |
| **Raw Logging** | optional | optional |
| **Sample queuing** | optional | required |

**Small queue may make sense for continuously powered configurations**
**Polling or file-based buffering makes sense for intermittently powered configurations**

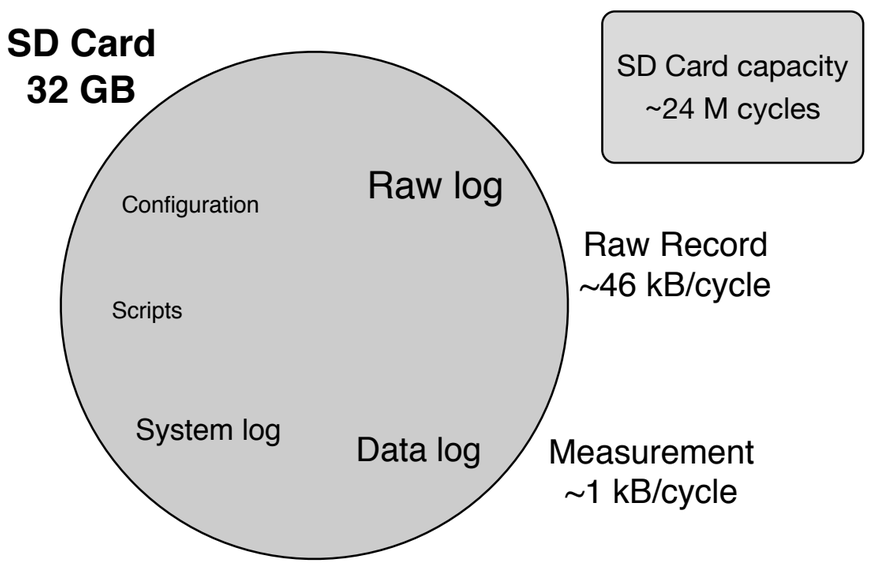



**Available RAM limits the amount of data that may be buffered**

Raw records are large (0.4 kB ea, ~46 kB/cycle)

Parsed samples are large compared to available RAM (~1 kB/cycle)

**RAM buffers don't persist across power cycles**




**SD Card access is slow compared to RAM**

- can read ~100 lines/s

Finding latest log entries is generally fast (near the beginning/end if file )

Random searches for data (e.g. by timestamp) are generally slow

li nmeas sets the output rate, then takes a series of samples.

Each sample is passed to the specified handler, along with a data structure that maintains measurement-specific state (varies per handler).

```
set bits li
scr delay 2000
scr macro ZERO 1000
li nmeas a 8 0.5 rr cz och1
scr macro SPAN 1000
li nmeas a 8 0.5 rr cs och1
scr macro AIR 1000
li nmeas a 5 0.5 rr ca och1
scr macro MEAS 1000
li nmeas b 10 0.5 rb cm och1
scr macro FILL 1000
li nmeas a 30 0.5 rs cm och1
li nmeas b 10 0.5 rb cm och1
scr macro PURGE 1000
clr bits p* v*
```

Optional parameter

**Measurement**

**LI NMEAS** handler | n | period | oflags | types | tmout

**Example -** `li nmeas a 60 0.5 ch`

**Diagnostic Measurement**

**LI DIAG** oflags | types

Pending

**Example -** `li diag ch`

Diagnostic measurement:
Content?
Context?

Licor raw

```
<li850>
 <data>
  <celltemp>2.37476e1</celltemp>
  <cellpres>1.01796e2</cellpres>
  <co2>4.98849e2</co2>
  <co2abs>1.0799157e-1</co2abs>
  <h2o>1.21172e1</h2o>
  <h2oabs>9.1537418e-2</h2oabs>
  <h2odewpoint>1.00119e1</h2odewpoint>
  <ivolt>1.2276172e1</ivolt>
  <raw>
    <co2>4403452</co2>
    <co2ref>4871240</co2ref>
    <h2o>2134376</h2o>
   <h2oref>3699180</h2oref>
  </raw>
  <flowrate>0</flowrate>
 </data>
 <ack>true</ack>
</li850>
```

licor_vrecord_t

```
uint64_t timestamp;
byte type;
byte ctx;
uint32_t seq_n;
uint32_t oflags;
double fields[FID_FIELD_COUNT];
```

avg_state_t

```
licor_xrecord_t data;
licor_vrecord_t M_prev;
licor_vrecord_t V_prev;
licor_vrecord_t S_prev;
```

bline_state_t

```
licor_brecord_t data;
double M_prev;
double S_prev;
```

licor_xrecord_t

```
uint64_t timestamps[2];
byte type;
byte ctx;
uint32_t seq_n;
uint16_t n;
licor_vrecord_t M;
licor_vrecord_t S;
```

bline_record_t

```
uint64_t timestamps[2];
byte type;
uint32_t seq_n;
byte ctx;
uint16_t n;
uint32_t oflags;
double epoints[2];
double M[2];
double S[2];
double min;
double max;
```

Data sizes

| | |
|---|---|
| byte | 1 |
| float | 4 |
| double | 4 |
| long double | 8 |
| long long | 8 |
| uint32_t | 4 |
| long | 4 |
| int | 2 |
| Time | 18 |
| licor_vrecord_t | 78 |
| licor_xrecord_t | 180 |
| licor_brecord_t | 52 |
| bline_state_t | 60 |
| avg_state_t | 414 |