Eco Net
Rustic OSSI

Technical Note

Version 1.2~~1~~
~~Septe~~November 2008

TOC

Introduction

This technical note presents the rustic OSSI: the interface over which one can control the *Eco Net* clusters in the praxis pilot implementation. The praxis itself is described in the "*Eco Net: Pilot*" paper.

The word *rustic* applies well also to this document, but we believe this presentation is adequate at the pilot phase of the project.

The OSSI is highly flexible; the format (and content selection) are configurable at compile time. There are two formats predefined: FMT_MAIN and FMT TERM settable in *options.sys*. The former the default, also used for Tcl/Tk scripts in the *Greenhouse Demo*. Here we used the latter, for clarity and easier to refer to multi-line output. Here is a sample of the difference:

FMT_MAIN (1 line):
```
1002   Agg 10 slot: 0, ts: 0.0:0:3  Col 104 slot: 0, ts: 0.0:0:0
S0: 0, S1: 0, S2: 0, S3: 0, S4: 1
```

FMT_TERM:
```
  Agg 10 slot: 0, ts: 0.0:0:3
  Col 104 slot: 0, ts: 0.0:0:0
  S0: 0, S1: 0, S2: 0, S3: 0, S4: 1
```

OSSI: Commands and output to OSS

Aggregator reports

Aggregators report to the master sensor readings from the collectors in their plots. The plot is defined by RF range, so a collector may be in several plots. If this built in redundancy is undesired, a simple registration mechanism can be added. The report format:

```
  Agg 10 slot: 5, time: 0.9:19:42
  Col 103 slot: 1, ts: 0.0:15:0
  S0: 1, S1: 2, S2: 3, S3: 4, S4: 0

  Agg 11 slot: 8, time: 0.12:21:38
  Col 116 slot: 2, time: 0.12:21:31
  S0: 0, S1: 7, S2: 7, S3: 7, S4: 7
```

line 1: Aggregator id, its nvm slot #, the reading time or timestamp on the agg.
line 2: Collector id, its slot #, the readings' time or timestamp on the collector
line 3: The sensors readings with customized labels, here S0 .. S4.

In the first version of the nodes (codenamed *Greenhouse Demo*), we had up to 5 sensors:

- (PAR) light via PAR,
- (T) temperature via the SHT sensor,

- (H) humidity via the SHT sensor,
- (PD) light via Si photodiode,
- (T2) temperature via the DS18B20.

In the second iteration destined for initial field deployments, PD was replaced by the EC-5 soil moisture sensor, and T2 is absent (S4 spot is reserved for a GPS module).

Technically, the single SHT provides two distinct readings, but for all purpose we should and will treat it as two distinct sensors (T, H).

Any sensor supported in PicOS can be attached, with its id encoded in the output (here S0, S1, S2, S3, S4). Raw readings are presented, easily convertible by OSS functionality.

Aggregator commands

This is the welcome screen:

```
EE from 0 to 16382 size 32
**EcoNet**1.1
Aggregator commands:
      Agg set / show:    a id [ audit_freq [ p_lev [ hex:a_fl ]]]
      Master Time:       T [ h:m:s ]
      Display data:      D [ from [ to [ col_id [ limit ]]]]
      Maintenance:       M (*** No collection until F     ***)
      Eprom erase:       E (*** deletes aggregated data   ***)
      Flash erase:       F (*** clears special conditions ***)
      Clean reset:       Q (*** to factory defaults (E+F) ***)
      ID set / show:     I[D id]    (*** CAREFUL Host ID    ***)
      ID master set:     IM id      (*** CAREFUL Master ID ***)
      Save(d) sys:       S[A]       (*** Show, SAve iFLASH ***)
      Sync coll:         Y [freq]   (*** Sync at freq      ***)
      Quit (reset)       q
      Help:              h
      Send master msg    m [ peg ]

      Col set / show:    c id agg_id [ Maj_freq [ min_freq [ rx_span
                                          [ hex:pl [ hex:c_fl]]]]]
      Find collector:    f col_id [ agg_id ]]
```

Optional trailing parameters can be omitted. Parameters that are to be ignored but are followed by significant ones should be entered as '*-1*' **with the exception of hex parameters** which must be either omitted or entered with proper values (-1 would be evaluated to 0x0001).

The first line shows EEPROM capacity:

```
EE from 0 to 16382 size 32
```

(EcoNet aggregator was compiled to use 16383 slots numbered from 0, 32 bytes each.)

EcoNet's version M.m[.x] signals the need of reflashing all nodes in the network; M.m must be consistent network-wide.

Command agg set / show:

`Agg set / show: a id [ audit_freq [ p_lev [ hex:a_fl ]]]`

Parameters:

audit_freq	Audit frequency, in seconds. Every that often the audit process checks the collectors known to the aggregator. Unconfirmed reports will be re-sent to the Master, collectors with missing two sensor readings will be marked and reported to the Master as '*gone*'. Set to 0, terminates the audit process. Range: [0, 2^16 -1], defaults to 13 seconds.
p_lev	Power level of the transmitter. CC1100 - specific. Range [0, 7], default 7. 0 translates to -20dBm, 7 to 10 dBm.
a_fl	Aggregator flags: bitmap of various binary switches. At present, only EEPROM-related flags are encoded: bit 0 set: collected (not confirmed) readings are written to EEPROM. bit 1 set: confirmed readings are written to EEPROM. bit 2 set: EEPROM is treated as a cyclical buffer (stored entries can be overwritten) **not implemented yet**. Range [0, 0xffff], defaults to 5.

The interpretation of **id** is:

absent (-1)	This aggregator, i.e., the one connected to the OSSI issuing the command
0	All aggregators, i.e., the command will be broadcast to the network
> 0	Aggregator id, i.e., the command refers to a specific aggregator, possibly remote

Examples:

`a`	Show data from the local (attached via OSSI) aggregator.

In most on-line cases scenarios the attached aggregator is the Master, i.e. the host controlling the cluster and aggregating data from all aggregators within.

In 'off-line' mode, any aggregator from the field can be attached to the OSSI and interrogated directly.

`a 11`	Show data from the aggregator with id 11.
`a -1 300`	Set audit freq. To 5 minutes on the local aggregator.

`a 0 -1 5` Set power level to 5 on all aggregators (may cut communication).

Note that the last (broadcast) variant is not reliable, and the values should be double checked on the aggregators for which there was no output data after this command. Similarly, there may be missing data after this '*broadcast show*':

`a 0`
whose **output** may look like this:

```
a 0
Stats for agg (BACA000A: 10):
 Audit freq 13 PLev 7 a_fl 0005 Uptime 24412 Mdelta 38 Master 10
 Stored entries 84 Mem free 4214 min 3900
Stats for agg (BACA000B: 11):
 Audit freq 13 PLev 7 a_fl 0005 Uptime 24412 Mdelta 24386 Master 10
 Stored entries 84 Mem free 4146 min 4032
```

Plev is power level. *Mdelta* (master delta) is the local uptime at the last network clock update, broadcast from the Master in periodic messages. *Stored entries* show how many sensor readings is stored in the NVM. Memory numbers show the RAM usage (in words) and dynamics. Only few of these data items are directly usable as the change confirmation, or for node monitoring. However, all can be used to assemble all sorts of alarm functionality. For example, a high water mark for NVM usage seems useful to trigger archiving and NVM clearing. Or, if only unconfirmed data are written to EEPROM, number of stored entries shows the end-to-end *ack* flow between a remote aggregator and the master.

The same output appears regardless whether the command is a set or show:

```
a -1 300
Stats for agg (BACA000A: 10):
 Audit freq 300 PLev 7 a_fl 0005 Uptime 1440 Mdelta 30 Master 10
 Stored entries 6 Mem free 4210 min 3932
```

Command master time:

```
Master Time:    T [ h:m:s ]
```

This command can be executed only on the attached Master. The time is propagated to all nodes in the cluster, and all sensor readings will be time-stamped using this time. If the time is not set, Master's boot time is propagated as the "0.0:0:0" time reference. For standalone plots without Masters, local boot times are used as the "0.0:0:0" time reference.

Note that the timestamps are not changed retroactively; however, this should be a simple OSS operation.

In the output, the timestamps using the Master's time are referred to as '*time*', otherwise they are '*ts*'.

Parameters:

h, m, s Three sane numbers for the hour, minutes, and seconds.

Must all be present (set), or all absent (show).

Examples:

`T`	Show attached Master's wall time.
`T 23:22:21`	Set attached Master's wall time.
`T 23 22 -21`	Same (the characters between numbers are ignored).

Sample output:

```
T
At ts 0.0:0:9 uptime 9

T 23:58:07
At time 0.23:58:7 uptime 24

T
At time 1.0:2:13 uptime 270
```

The leading number (**1**.0:2:13) is the number of days, extending the clock without a calendar.

Command display data:

`Display data:   D [ from [ to [ col_id [ limit ]]]]`

This command may search through large nvm devices and produce large output, potentially interleaved with reports from other aggregators or output from other commands. To avoid overwhelming UART and starving other functionality, the output and traverse through NVM is throttled down to about four entries per second. Although *from*, *to* are optimized to the known NVM content, large ranges should be avoided when the attached aggregator is performing its duties on-line.

Note that '*from*' and 'to*'* can describe descending order, useful for tracking back missing sensor readings. **'-1' don't apply here**; either true values or nothing.

Parameters:

from	Slot number to begin the search for data (inclusively). Range: [0, 2^32 -1], defaults to the beginning of NVM for the readings.
to	Slot number to end the search (inclusively). Range: [0, 2^32 -1], defaults the end of NVM for the readings.
col_id	Collector node Id (only the data for this collector will be displayed) or 0 for 'all'.
limit	Optional restriction on the number of lines to be displayed. Range: [0, 2^16 -1], default 0 (no limit).

Examples:

`D`	Display all stored data.
`D 10 0 102 4`	Display up to 4 coll. 102 readings from the 10th slot down.

Commands M, E, F, Q, I, S, Y, q, h:

```
Maintenance:    M (*** No collection until F     ***)
Eprom erase:    E (*** deletes aggregated data   ***)
Flash erase:    F (*** clears special conditions ***)
Clean reset:    Q (*** to factory defaults (E+F) ***)
ID set / show:  I[D id]    (*** CAREFUL Host ID    ***)
ID master set:  IM id      (*** CAREFUL Master ID ***)
Save(d) sys:    S[A]       (*** Show, SAve iFLASH ***)
Sync coll:      Y [freq]   (*** Sync at freq      ***)
Quit (reset)    q
Help:           h
```

M	For maintenance only (e.g. direct EEPROM dumps).
E	Warning: no mercy prompt. Takes about 15 seconds.
F	Warning: no mercy prompt.
Q	Warning: no mercy prompt. Takes about 15 seconds.
I	Show Ids (all stats data)
ID <id>	Set node id to <id>, shaw all stats data
IM <id>	Set Master id to <id>, shaw all data
S	Show sys param in iFLASH
SA	Save sys param to iFLASH
Y <freq>	Synchronize all collectors to <freq>; offset from midnight of the last T command
q	Reset
h	Displays the menu (intro screen).

The node is reset after the M, E, F, Q, q operations. In the field, Q **is planned** to be implemented on the '*long reset*' button and '*soft reset*' is F.

ID, IM should be used only by conscious operators, as side effects may be dreadful. Let's illustrate the other commands by the full flow leading to a saved setup of the master enforcing synchronized collections every minute:

Agg 10 slot: 0, ts: 0.0:0:10 Col 102 slot: 0, ts: 0.0:0:4 S0: 0, S1: 0, S2: 0, S3: 0, S4: 1
Agg 10 slot: 0, ts: 0.0:0:14 Col 101 slot: 0, ts: 0.0:0:4 S0: 0, S1: 0, S2: 0, S3: 0, S4: 1
Agg 10 slot: 0, ts: 0.0:0:39 Col 101 slot: 0, ts: 0.0:0:34 S0: 0, S1: 0, S2: 0, S3: 0, S4: 2
Agg 10 slot: 0, ts: 0.0:0:40 Col 102 slot: 0, ts: 0.0:0:34 S0: 0, S1: 0, S2: 0, S3: 0, S4: 2

[Regular stuff: collectors 101, 102 report to Master 10 every 30 s. Let's set the time:]

T 22:05:11
At time 0.22:5:11 uptime 52

[Synchronize collectors to 30 s:]

Y 30
Synced to 30

Agg 10 slot: 0, time: 0.22:5:26 Col 102 slot: 0, ts: 0.0:1:4 S0: 0, S1: 0, S2: 0, S3: 0, S4: 3
Agg 10 slot: 0, time: 0.22:5:29 Col 101 slot: 0, ts: 0.0:1:4 S0: 0, S1: 0, S2: 0, S3: 0, S4: 3

[the above: agg. time is updated, coll. not yet; the below: time is updated on the coll, and synchronized to 30 s.]

Agg 10 slot: 0, time: 0.22:6:2 Col 101 slot: 0, time: 0.22:6:0 S0: 0, S1: 0, S2: 0, S3: 0, S4: 4
Agg 10 slot: 0, time: 0.22:6:5 Col 102 slot: 0, time: 0.22:6:0 S0: 0, S1: 0, S2: 0, S3: 0, S4: 4
Agg 10 slot: 0, time: 0.22:6:34 Col 101 slot: 0, time: 0.22:6:30 S0: 0, S1: 0, S2: 0, S3: 0, S4: 5
Agg 10 slot: 0, time: 0.22:6:35 Col 102 slot: 0, time: 0.22:6:30 S0: 0, S1: 0, S2: 0, S3: 0, S4: 5
Agg 10 slot: 0, time: 0.22:7:2 Col 102 slot: 0, time: 0.22:7:0 S0: 0, S1: 0, S2: 0, S3: 0, S4: 6
Agg 10 slot: 0, time: 0.22:7:4 Col 101 slot: 0, time: 0.22:7:0 S0: 0, S1: 0, S2: 0, S3: 0, S4: 6

a 10 -1 -1 3
Stats for agg (BACA000A: 10): Audit freq 13 PLev 7 a_fl 0003 Uptime 185 Mdelta 52 Master 10
Stored entries 0 Mem free 4176 min 3746

[We intend to detach the aggregator, turning it into data logger, so we changed the flags to store all the data.]

SA
Flash: id 10 pl 7 a_fl 002C au_fr 13 master 10 sync_fr 30

[Saved sys data.]

Agg 10 slot: 0, time: 0.22:7:33 Col 102 slot: 0, time: 0.22:7:30 S0: 0, S1: 0, S2: 0, S3: 0, S4: 7
Agg 10 slot: 1, time: 0.22:7:40 Col 101 slot: 0, time: 0.22:7:30 S0: 0, S1: 0, S2: 0, S3: 0, S4: 7

[Note 'slot 1': sensor data are stored even if they're 'confirmed' at the Master.]

q

(welcome menu)

Implicit T 0:0:0
Agg 10 slot: 3, time: 0.0:0:3 Col 101 slot: 0, time: 0.22:8:0 S0: 0, S1: 0, S2: 0, S3: 0, S4: 8
Agg 10 slot: 2, time: 0.0:0:3 Col 102 slot: 0, time: 0.22:8:0 S0: 0, S1: 0, S2: 0, S3: 0, S4: 8
Agg 10 slot: 4, time: 0.0:0:32 Col 101 slot: 0, time: 0.0:0:30 S0: 0, S1: 0, S2: 0, S3: 0, S4: 9
Agg 10 slot: 5, time: 0.0:0:35 Col 102 slot: 0, time: 0.0:0:30 S0: 0, S1: 0, S2: 0, S3: 0, S4: 9
Agg 10 slot: 6, time: 0.0:1:3 Col 101 slot: 0, time: 0.0:1:0 S0: 0, S1: 0, S2: 0, S3: 0, S4: 10
Agg 10 slot: 7, time: 0.0:1:6 Col 102 slot: 0, time: 0.0:1:0 S0: 0, S1: 0, S2: 0, S3: 0, S4: 10

[Saved synchronization frequency caused implicit time setup to midnight at startup. A new T command would adjust the wall time, and consecutive collections would be synchronized to the new clock. Synchronization changes the collection frequency stored on the collectors, so they keep synchronized even if the Master is switched off. Y 0 switches the synchronization off.]

Command collector set / show:

```
Col set / show: c id agg_id [ Maj_freq [ min_freq [ rx_span
                                                  [ hex:pl [ hex:c_fl]]]]]
```

This command is allowed only on the Master. It sets or retrieves values of the main parameters from the collector (*id*), via the aggregator (*agg_id*). The result is displayed on the Master.

Note that the collectors are not part of the network, designed to be in low power mode most of the time. The request is forwarded to the (*agg_id*) and cached there until the collector (*id*) wakes up to read sensors and report to the aggregators in proximity. At that time, it also instructs the aggregators that it is ready to accept cached messages, along with the acks for sensor data. Then, the aggregator forwards the cached message, receives the response, and forwards it to the Master.

In the present version, only one (last) outstanding request is cached in the forwarding aggregator.

Various combinations and special values of the parameters provide enough flexibility for exotic deployments (standalone collectors, different duty cycles and collection schedules), but we don't explore them here.

Parameters:

id	Collector id of the subject.
agg_id	The aggregator in the collector's proximity, to store and forward the request. Zero or lack of it means the local aggregator (Master).
Maj_freq	The sensor reading and reporting frequency, in seconds. Range: [0, 2^16 -1]. Defaults to 900. 0 switches the collecting and reporting off.
min_freq	The frequency with which the sensor readings are resent if not confirmed by an aggregator, in seconds. Can be thought of as timeout for retries with some additional functionality. Range: [0, 2^16 -1]. Defaults to 5. Functionally insane values are ignored.
rx_span	Number of milliseconds the collector opens its Rx for, after transmissions. 0 - Rx always off, 1 - Rx always on. Range: [0, 2^16 -1], default 1 (always on). This value may significantly impact further communication with the collector.
pl_vec	Hexadecimal number representing power levels for up to 4 retries. This general functionality may play role also in the Eco Net praxis, for fine tuning power management. For the demonstration, it may be useful to set optimal RF ranges for desired sensor grouping. Range [0000, ffff], defaults to 7777 (four attempts at power level 7 each).
c_fl	At present the same as a_fl flags in the 'a' command.

Examples:

`c 101` Display the numbers from the collector 101 within the Master's range.

`c 101 0 60` Set collecting and reporting frequency to 60 seconds on the collector 101 off the Master.

`C 202 11 -1 -1 1 7531` On the collector 202 in the agg. 11 plot:
permanently open rx,
set the transmitter to power level 1,
if necessary send retries on 3, at 5, and finally at 7.

Note that the new values take effect after the current cycle ends. For example, if the reporting frequency changes from 15 minutes to 30 seconds, it'll happen after the end of the present 15 min sleep period.

Side effects of a parameter change may be not that obvious. For example, after extending *Maj_freq* on a collector, the audit process on an aggregator may report the collector as '*gone*'.

Command find collector:

`Find collector: f col_id [ agg_id ]]`

Parameters:

id	Collector to find. 0 – ***count*** *all*.
agg_id	The aggregator to query. 0 - all (broadcast).

Output:

```
f
f 0 0
Agg 10 handles 3 collectors
Agg 11 handles 3 collectors

f 102
  Agg 10 slot: 15, time: 0.3:39:59
  Col 102 slot: 12, time: 0.3:39:58
  PAR: 0, T: 0, H: 0, PD: 0, T2: 0

f 117
  Agg 11 slot: 5, time: 0.3:41:21
  Col 117 slot: 2, ts: 0.0:30:1
  PAR: 0, T: 0, H: 0, PD: 0, T2: 0

f 208
  Agg 10 slot: 20, time: 0.3:44:35
  Col 208 slot: 0, ts: 0.0:0:0
  PAR: 0, T: 0, H: 0, PD: 0, T2: 0

  Agg 11 slot: 8, time: 0.3:44:35
```

```
Col 208 slot: 0, ts: 0.0:0:0
PAR: 0, T: 0, H: 0, PD: 0, T2: 0
```

Collector commands

```
EE from 0 to 32766 size 16
***EcoNet***
Collector commands
        Set/ show:      s [ Maj_freq [ min_freq [ rx_span
                                    [ hex:pl_vec [ hex:c_fl ]]]]]
        Display data:   D [ from [ to [ status [ limit ]]]]
        Maintenance:    M (*** No collection until F      ***)
        Eprom erase:    E (*** deletes all collected data ***)
        Flash erase:    F (*** clears special conditions  ***)
        Clean reset:    Q (*** to factory defaults (E+F)  ***)
        ID set / show:  I[D id]     (*** CAREFUL Host ID    ***)
        Save(d) sys:    S[A]        (*** Show, SAve iFLASH ***)
        Quit (reset)    q
        Help:           h
```

Note the fundamental difference between the aggregator and collector OSSI: while the aggregator OSSI is part of the regular operations on any network, on- or off-line, the collector interface is operational only off-line, or on-line with a standalone collector only.

In this praxis version execution of collector commands triggered remotely is blocked, and only necessary functionality of set / show is triggered via 'c' command on the aggregator. The same approach is likely to extend on the final praxis, as the collectors are not truly part of the network, and under severe power budget restrictions.

The first line shows EEPROM capacity:

`EE from 0 to 32766 size 16`
(EcoNet collector was compiled to use 32766 slots numbered from 0, 16 bytes each.)

Command set / show:

```
Set/ show:      s [ freq_Maj [ freq_min [ rx_span [ hex:pl_vec
                                                  [ hex:c_fl ]]]]]
```

See '*c*' command in the Aggregator section.

Command display data:

`Display data:   D [ from [ to [ status [ limit ]]]]`

See '*D*' command in the Aggregator section, with '*col_id*' replaced with '*status*'.

status Selector on the value of *status*:
0 - all (default)

1 - confirmed (by an aggregator)
2 - collected (but not confirmed)
3 - in use (being collected)
other - all

Examples:

D 40 20 2 Lists entries from the 40th down to the 20th slot, collected but not confirmed by any aggregator.

Output:

```
D
CONFIRMED slot 0 ts 0.0:0:0: S0: 0, S1: 1, S2: 1, S3: 1, S4: 1
CONFIRMED slot 1 ts 0.0:0:0: S0: 0, S1: 2, S2: 2, S3: 2, S4: 2
COLLECTED slot 2 ts 0.0:15:0: S0: 0, S1: 3, S2: 3, S3: 3, S4: 3
COLLECTED slot 3 ts 0.0:15:31: S0: 0, S1: 4, S2: 4, S3: 4, S4: 4
Collector 102 direct dump: slots 0 -> 4 status ALL upto 0 #5
COLLECTED slot 4 ts 0.0:16:1: S0: 0, S1: 5, S2: 5, S3: 5, S4: 5
```

The node was reset after the 1st reading, and after the 2nd the Maj_freq was changed to 30 seconds. The #5 at the closing line is the total number of entries listed.

```
D 100000 0 2
COLLECTED slot 6 ts 0.0:17:2: S0: 0, S1: 7, S2: 7, S3: 7, S4: 7
COLLECTED slot 5 ts 0.0:16:32: S0: 0, S1: 6, S2: 6, S3: 6, S4: 6
COLLECTED slot 4 ts 0.0:16:1: S0: 0, S1: 5, S2: 5, S3: 5, S4: 5
COLLECTED slot 3 ts 0.0:15:31: S0: 0, S1: 4, S2: 4, S3: 4, S4: 4
COLLECTED slot 2 ts 0.0:15:0: S0: 3, S1: 3, S2: 3, S3: 3, S4: 3
Collector 102 direct dump: slots 6 -> 0 status COLLECTED upto 0 #5
```

Note the occasional line swap. This is a side effect of the output queuing. It is left unattended as harmless.