

#####OMC-048_FW_0.01B1268_H0002.bin#####

Added OMC-2900 serial output

By defining an output driver in the config file, it is possible to convert sensor data to an OMC-2900 output over a serial port. The description on how-to-use is provided in the OMC2900.py file located in the script/modules folder.

Minor patches / fixes:

- *The EXO and ysi_6_series parameter tables are corrected on tag names/units.*
- *The status indicator of the log format records is reworked to match the omc045III style.*
- *In case a sensor is timed out a record is logged containing an invalid data status indication.*

#####OMC-048_FW_0.01B1228_H0002.bin#####

Added NTP time synchro

The Time can now be synchronized via the NTP protocol. It can be enabled via 'utc_time_sync'. And optionally the server and port can be provided by the user (defaults: europe.pool.ntp.org, port: 123)

```
# ----FTP-Settings---- #  
Ftp:  
- id: data_ftp  
  utc_time_sync: True  
  utc_time_server: nl.pool.ntp.org  
  utc_time_server_port: 123
```

The NTP server is checked after each scheduled FTP task if enabled. After the transmission of the FTP files the NTP server is requested. After a valid NTP package was received the time and date is corrected (taking the loggers UTC offset into account!). To prevent numerous requests to the NTP server a 4 hour timeout was introduced. This prevents unwanted and unnecessary requests to the server.

During startup:

```
2020-11-27 15:54:22 [FTP] INFO FTP configuration detected: ftp.omc-data-online.com  
2020-11-27 15:54:22 [NTP] INFO Automatic NTP time synchronisation enabled on data_ftp  
2020-11-27 15:54:22 [NTP] INFO address: nl.pool.ntp.org port: 123
```

During normal operation:

```
2020-11-27 16:00:48 [NTP] INFO NTP server check initiated  
2020-11-27 16:00:48 [NTP] INFO Logger time deviation to NTP source > 2 second  
2020-11-27 17:00:49 [NTP] INFO Logger time synchronised to NTP source
```

or in case of earlier NTP check:

```
2020-11-27 17:10:47 [NTP] INFO NTP server check skipped, last check was less then 4 hours ago
```

Added NMEA PTID driver:

The nmea PTID driver is a static driver designed for the Orinoco solo tide sensor. The driver is NMEA based and requires the static input format as:

\$PTID,005,10072006,092114,0.08,M,00,12.36,V,1023,hPa,x*19

The parameters which are obtained are: **Depth**, **Status**, **Battery voltage** and **Barometric pressure**. This driver is statically created due to the absence of a 'generic nmea' driver at this moment.

Note: The sensor transmits data at the interval of 2 minutes. It is important to make sure the logger is awake when the sample is transmitted! Therefore the sample interval should be set to 2 minutes

```
nmea_ptid:
- id: ptid0
  port: serial1
  mode: RS232
  baudrate: 9600
  sample_interval: "0 0,2,4,6,8,10,12,14,16,18,20,22,24,26,28,30,32,34,36,38,40,42,44,46,48,50,52,54,56,58 * * *"
  response_timeout: 125
```

and the timeout to more than 2 minutes! Also make note that the startup procedure can be 'delayed' by up to 2 minutes due to the slow data response of the sensor! See example below:

Changes navilock_md6.py:

A minor change in the collection of the nmea data transmitted by the navilock sensor. In case of slower data outputs the nmea sentences were parsed with prior to last recent data.

Software patches / fixes:

- Updated to micropython 1.13
- *read_line.py* refactored using async stream, due to harfault uart serial problems. The *update()* now returns the list of sentences, the *get()* is removed
- All Sensor drivers refactored to comply to new *read_line*.
- PAR parameters added to the EXO table, these were somehow forgotten.
- ATparser/modem refactored to work with streamreader
- Errors on corrupt transmitted directory or unreadable files now delete the 'problematic files' instead of trying to re-send.
- In *driver_manager*: In case a sensor times-out the power (if assigned) is force toggled with a off-time given in the sensor driver (default 5 sec).
- Some minor changes in the syslog levels to prevent spamming of data or unnecessary messages to the user (such as 'wake up message', and 'usb_wake')
- In *navilock*: The datetime sync (if enabled) is immediately executed after the receipt of the RMC message, this prevents the possibility the time could be synced to an 'older' RMC message. In case a logger lost its time the datetime resets to 2000-1-1-1, which was also the date the navilock compared to the first cycle. This caused the datetime not to be updated for 4 hours, changed the starting date to the not-reset date of the logger.
- Ysi-6-series: Fix on the temperature bug, multiple spaces as divider are now possible and does not cause the sensor to reject the data message.
- Added the *dual_protocol()* function to USB which can change between REPL only and MSC+REPL. This is later to be used for the usb/sd problem fix. This also works for connecting and reconnecting usb during sleep/wake
- The scheduler sometime executed tasks too early. The cron interval is now checked prior to executing the task. If the cron interval is not yet due it shall go to sleep again for the remainder of the time.

#####OMC-048_FW_0.01B1038_H0002.bin#####

Added navilock_md6 GPS time synchronisation:

The GPS time can be automatically synced on each valid sample that is received. It is enabled/disabled via 'utc_time_sync:' which is driver specific (default is disabled). info messages are reported if the time is synced.

```
Gps:
- id: GPS_A
  utc_time_sync: True
```

Added UTC offset in configuration:

The system tab can now contain 'utc_time_offset_hours:' and 'utc_time_offset_minutes:' (zero by default). Positive or negative offsets are given via +x (or x) and -x.

```
# ----System---- #
Omc048:
  utc_time_offset_hours: +1
  utc_time_offset_minutes: -30
```

EXO driver wiper:

EXO driver wiper implementation changed from 'wipe_after_number_of_samples:' to 'samples_per_wipe:'. '0' means no wipes, '1' means wipe each sample, '10' means wipe first sample, don't wipe the next 9 samples. etc.

The wipe is executed before a measurement(sample), this shall delay the data request with the wipe time (around 1 min)

#####OMC-048_FW_0.01B990_H0002.bin#####

Added Usb_wake:

Usb wake is used to periodically wake up the USB interface to check if a usb connection is active. If the usb is found the log file is stopped prior and started after usb negotiations, this reduces the chance of corrupt file detection by windows.

In config.txt:

```
Usb_wake:
- id: usb_wakeup
  wake_interval: "0,10,20,30,40,50 * * * *"
```

Added the detection of USB wake, FTP and Data interval on start-up info messaging:

In repl:

```
2020-10-29 12:22:39 [LOGGER_MANAGER] INFO Data file create interval detected'
2020-10-29 12:22:39 [FTP] INFO FTP configuration detected: ftp.omc-data-online.com'
2020-10-29 12:22:39 [USB_WAKE] INFO Usb wake interval configuration detected'
```

