

libais for CRC and AISHub.net NMEA 4.10 data

This project is a modified version of [libais](#).

Following modifications were made so it can parse NMEA and NMEA 4.10 data from the CRC local antenna and from the AISHub feed:

- Allow messages without station (\s field in NMEA 4.10 tag).
- Allow messages without timestamp (\c field in NMEA 4.10 tag).
- Group multi-part messages by field \g in NMEA 4.10 tag and sequence ID.
In the original libais code, multi-part messages were grouped using the tuple (\s field in NMEA 4.10 tag, AIS sequence ID) (2954,7 in the following exemple).

The reason for these changes is that multi-part messages does not have these fields in there tagblock except for the first one. Example:

```
\g:1-2-22890,c:1659589714,s:2954*52\!AIVDM,2,1,7,B,55Uilt800000cQW?
07Lnpbq@4dpn2222222220e3`;525wVN3l3lU80hCU3,0*20
\g:2-2-22890*46\!AIVDM,2,2,7,B,s0hCSQDp880,2*7E
```

The modified code is the following (in *ais/stream/_init_.py*):

```
def normalize(...
    allow_unknown=True,
    ...
    allow_missing_timestamps=True,
    ....
):
```

Replacing line 234:

```
bufferSlot = (tagblock_station, station, fields[3])
```

by (use \g NMEA 4.10 tag field instead of the previous tuple):

```
try:
    bufferSlot = (tagblock['tagblock_group']['id'], fields[3])
except KeyError:
    bufferSlot = (fields[3],)
```

Installation

```
$ python setup.py build
$ python setup.py install
```

Usage/Examples

Usage

To parse a file containing NMEA messages, the following code can be used:

```
import ais.stream
with open("ais_sample") as f:
    for msg in ais.stream.decode(f):
        print(msg)
```

The variable *msg* contains a dictionary with parsed AIS message fields.

Example:

An implementation example is given in *example.py*.

ais_sample :

```
\s:3392,c:1653429872*34\!AIVDM,1,1,,A,13BAIL00000W=jJN`sa9VDgp0>`<,0*0A
\s:3235,c:1653429872*38\!AIVDM,1,1,,B,ENk`spD973h9@6:@@@@@@@@@@@@=3DAN7w?
0800003vP000,0*55
\g:1-2-
820068245,s:3179,c:1653429872*76\!AIVDM,2,1,3,A,53@pPup2GAFTu`TF220PE8lE>22222222
222220l20>846inN=U3mjCQ,0*31
\g:2-2-820068245*6A\!AIVDM,2,2,3,A,p2C`0@DR5Dp8880,2*19
\s:2639,c:1653429872*31\!AIVDM,1,1,,B,B52gvb@00=qrKhTcLWCQ3wh1nDLr,0*79
\s:3031,c:1653429872*3E\!AIVDM,1,1,,B,B6:`lg@09J9TE1TTLbRaWwf6SP06,0*72
\g:1-2-820068246,s:3031,c:1653429872*78\!AIVDM,2,1,5,B,5<3<tJ2`h;6S<Bck:RS?2Sk?
2Sk?2Sk?2Sk?2Shn6HkkkP000AL:??<;=:e=C,0*1C
\g:2-2-820068246*69\!AIVDM,2,2,5,B,BP0000000:P,2*5A
```

Running *example.py* :

```
$ python example.py ais_sample
```

Output on console:

```
{'id': 1, 'repeat_indicator': 0, 'mmsi': 220486000, 'nav_status': 0,
'rot_over_range': False, 'rot': 0.0, 'sog': 0.0, 'position_accuracy': 0, 'x':
8.566741666666667, 'y': 53.54651333333333, 'cog': 245.6999969482422,
'true_heading': 151, 'timestamp': 60, 'special_manoeuvre': 0, 'spare': 0, 'raim':
False, 'sync_state': 0, 'slot_timeout': 3, 'received_stations': 10764,
'tagblock_station': '3392', 'tagblock_timestamp': 1653429872}

{'id': 21, 'repeat_indicator': 1, 'mmsi': 993672161, 'spare': 0, 'aton_type': 8,
'name': 'RNG R LT @', 'position_accuracy': 0, 'x': -82.43696333333334,
'y': 27.920215, 'dim_a': 0, 'dim_b': 0, 'dim_c': 0, 'dim_d': 0, 'fix_type': 7,
'timestamp': 61, 'off_pos': False, 'aton_status': 0, 'raim': False,
'virtual_aton': False, 'assigned_mode': False, 'tagblock_station': '3235',
'tagblock_timestamp': 1653429872}

{'id': 5, 'repeat_indicator': 0, 'mmsi': 219029751, 'ais_version': 2, 'imo_num':
9913705, 'callsign': 'OZIE ', 'name': 'HERMES ', 'type_and_cargo':
52, 'dim_a': 16, 'dim_b': 14, 'dim_c': 8, 'dim_d': 4, 'fix_type': 1, 'eta_month':
11, 'eta_day': 3, 'eta_hour': 22, 'eta_minute': 30, 'draught': 5.400000095367432,
'destination': 'TOWING IN AARHUS ', 'dte': 0, 'spare': 0, 'tagblock_group':
{'sentence': 1, 'groupsize': 2, 'id': 820068245}, 'tagblock_station': '3179',
'tagblock_timestamp': 1653429872}

{'id': 18, 'repeat_indicator': 0, 'mmsi': 338427561, 'spare': 0, 'sog': 0.0,
'position_accuracy': 1, 'x': -117.166985, 'y': 32.707606666666666, 'cog': 360.0,
'true_heading': 511, 'timestamp': 32, 'spare2': 0, 'unit_flag': 0,
'display_flag': 0, 'dsc_flag': 1, 'band_flag': 1, 'm22_flag': 1, 'mode_flag': 0,
'raim': True, 'commstate_flag': 1, 'slot_increment': 5235, 'slots_to_allocate':
5, 'keep_flag': False, 'tagblock_station': '2639', 'tagblock_timestamp':
1653429872}

{'id': 18, 'repeat_indicator': 0, 'mmsi': 413807805, 'spare': 0, 'sog':
3.700000047683716, 'position_accuracy': 1, 'x': 120.20843166666667, 'y':
31.943106666666665, 'cog': 271.29998779296875, 'true_heading': 511, 'timestamp':
28, 'spare2': 0, 'unit_flag': 1, 'display_flag': 1, 'dsc_flag': 0, 'band_flag':
1, 'm22_flag': 0, 'mode_flag': 0, 'raim': False, 'commstate_flag': 1,
'commstate_cs_fill': 393222, 'tagblock_station': '3031', 'tagblock_timestamp':
1653429872}

{'id': 5, 'repeat_indicator': 0, 'mmsi': 808664168, 'ais_version': 0, 'imo_num':
707800168, 'callsign': '3D$<2(', 'name': '30(<30(<30(<30(<30(<',
'type_and_cargo': 54, 'dim_a': 51, 'dim_b': 51, 'dim_c': 51, 'dim_d': 51,
'fix_type': 8, 'eta_month': 0, 'eta_day': 0, 'eta_hour': 0, 'eta_minute': 0,
'draught': 6.900000095367432, 'destination': '0(<0,4*45MJ@@@@@*', 'dte': 0,
'spare': 0, 'tagblock_group': {'sentence': 1, 'groupsize': 2, 'id': 820068246},
'tagblock_station': '3031', 'tagblock_timestamp': 1653429872}
```

Used By

This project is used by Mines Paris CRC (<https://www.crc.mines-paristech.fr/fr/>).

Authors

- Ambroise Renaud (ambroise.renaud@minesparis.psl.eu)

