```
--== PHY ==--

868.2Mhz
Syncword 0xF1
250kHz Bandwidth
Spreading Factor 7
ExplicitHeader: Coding Rate CR 5-8/8 (depending on #neighbors), CRC for Payload

--== FANET MAC ==--


Header:
[Byte 0]
7bit            Extended Header
6bit            Forward
5-0bit          Type

Source Address:
[Byte 1-3]
1byte           Manufacturer
2byte           Unique ID (Little Endian)

Extended Header:
[Byte 4 (if Extended Header bit is set)]
7-6bit          ACK:
                        0: none (default)
                        1: requested
                        2: requested (via forward, if received via forward (received forward bit = 0). must be used if
forward is set)
                        3: reserved
5bit            Cast:
                        0: Broadcast (default)
                        1: Unicast (adds destination address (8+16bit)) (shall only be forwarded if dest addr in cache
and no 'better' retransmission received)
4bit            Signature (if 1, add 4byte)
3bit            Geo-based Forwarded     (prevent any further geo-based forwarding, can be ignored by any none-
forwarding instances)
2-0bit          Reserved        (ideas: indicate multicast interest add 16bit addr, emergency)

Destination Address (if unicast is set):
[Byte 5-7]
1byte           Manufacturer
2byte           Unique ID (Little Endian)

Signature (if signature bit is set):
[Byte 5-8 or Byte 8-11 (if unicast is set)]
4byte           Signature



Types:
-----------

ACK (Type = 0)
No Payload, must be unicast

-----------

Tracking (Type = 1)
[recommended intervall: floor((#neighbors/10 + 1) * 5s) ]
Note: Done by app layer of the fanet module

[Byte 0-2]      Position        (Little Endian, 2-Complement)
bit 0-23        Latitude        (Absolute, see below)
[Byte 3-5]      Position        (Little Endian, 2-Complement)
bit 0-23        Longitude       (Absolute, see below)

[Byte 6-7]      Type            (Little Endian)
bit 15          Online Tracking
bit 12-14       Aircraft Type
                        0: Other
                        1: Paraglider
                        2: Hangglider
                        3: Balloon
                        4: Glider
                        5: Powered Aircraft
                        6: Helicopter
                        7: UAV
bit 11          Altitude Scaling 1->4x, 0->1x
bit 0-10        Altitude in m

[Byte 8]        Speed           (max 317.5km/h)
bit 7           Scaling         1->5x, 0->1x
bit 0-6         Value           in 0.5km/h

[Byte 9]        Climb           (max +/- 31.5m/s, 2-Complement)
bit 7           Scaling         1->5x, 0->1x
bit 0-6         Value           in 0.1m/s

[Byte 10]       Heading
bit 0-7         Value           in 360/256 deg
```

```
[optional]
[Byte 11]       Turn rate       (max +/- 64deg/s, positive is clock wise, 2-Complement)
bit 7           Scaling         1->4x, 0->1x
bit 0-6         Value           in 0.25deg/s

[optional, if used byte 11 is mandatory as well]
[Byte 12]       QNE offset      (=QNE-GPS altitude, max +/- 254m, 2-Complement)
bit 7           Scaling         1->4x, 0->1x
bit 0-6         Value           in m

------------

Name (Type = 2)
[recommended intervall: every 4min]

8bit String (of arbitrary length, \0 termination not required)

------------

Message (Type = 3)

[Byte 0]        Header
bit 0-7         Subheader, Subtype (TBD)
                        0: Normal Message

8bit String (of arbitrary length)

------------

Service (Type = 4)
[recommended intervall: 40sec]

[Byte 0]        Header  (additional payload will be added in order 6 to 1, followed by Extended Header payload 7 to 0
once defined)
bit 7           Internet Gateway (no additional payload required, other than a position)
bit 6           Temperature (+1byte in 0.5 degree, 2-Complement)
bit 5           Wind (+3byte: 1byte Heading in 360/256 degree, 1byte speed and 1byte gusts in 0.2km/h (each: bit 7
scale 5x or 1x, bit 0-6))
bit 4           Humidity (+1byte: in 0.4% (%rh*10/4))
bit 3           Barometric pressure normailized (+2byte: in 10Pa, offset by 430hPa, unsigned little endian (hPa-
430)*10)
bit 2           Support for Remote Configuration (Advertisement)
bit 1           State of Charge  (+1byte lower 4 bits: 0x00 = 0%, 0x01 = 6.666%, .. 0x0F = 100%)
bit 0           Extended Header (+1byte directly after byte 0)
                The following is only mandatory if no additional data will be added. Broadcasting only the
gateway/remote-cfg flag doesn't require pos information.
[Byte 1-3 or Byte 2-4]  Position        (Little Endian, 2-Complement)
bit 0-23        Latitude        (Absolute, see below)
[Byte 4-6 or Byte 5-7]  Position        (Little Endian, 2-Complement)
bit 0-23        Longitude   (Absolute, see below)
+ additional data according to the sub header order (bit 6 down to 1)

------------

Landmarks (Type = 5)
Note: Landmarks are completely independent. Thus the first coordinate in each packet has to be an absolute one. All
others are compressed in relation to the one before.
Note2: Identification/detection shall be done by hashing the whole payload, excluding bytes 0, 1 and, 2 (optional).
That way one quietly can change the layer to 'Don't care' and quickly
        destroy the landmark w/o having to wait for it's relative live span to be exceeded.
Note3: In case a text has the same postion as the first position of any other landmark then the text is considered to
be the label of that landmark.

[Byte 0]
bit 4-7         Time to live +1 in 10min (bit 7 scale 6x or 1x, bit 4-6) (0->10min, 1->20min, ..., F->8h)
bit 0-3         Subtype:
                        0:      Text
                        1:      Line
                        2:      Arrow
                        3:      Area
                        4:      Area Filled
                        5:      Circle
                        6:      Circle Filled
                        7:      3D Line          suitable for cables
                        8:      3D Area          suitable for airspaces (filled if starts from GND=0)
                        9:      3D Cylinder      suitable for airspaces (filled if starts from GND=0)
                        10-15: TBD
[Byte 1]
bit 7-5         Reserved
bit 4           Internal wind dependency (+1byte wind sector)
bit 3-0         Layer:
                        0:      Info
                        1:      Warning
                        2:      Keep out
                        3:      Touch down
                        4:      No airspace warn zone            (not yet implemented)
                        5-14:   TBD
                        15:     Don't care
[Byte 2 only if internal wind bit is set] Wind sectors +/-22.5degree (only display landmark if internal wind is within
one of the advertised sectors.
                                                                        If byte 2 is present but is zero, landmark gets
only displayed in case of no wind)
```

```
bit 7           NW
bit 6           W
bit 5           SW
bit 4           S
bit 3           SE
bit 2           E
bit 1           NE
bit 0           N

[n Elements]
                        Text (0):               Position (Absolute) + String                            //(2 Byte
aligned, zero-termination is optional)
                        Line/Arrow (1,2):       Position (1st absolute others compressed, see below, minimum 2
elements)
                        Area (filled)(3,4):     Position (1st absolute others compressed, see below, minimum 3
elements)
                        Circle (filled)(5,6):   n times: Position (1st absolute others compressed, see below) + Radius
(1Byte in 50m, bit 7 scale 8x or 1x, bit 0-6)
                        3D Line (7):            n times: Position (1st in packet absolute others compressed, see below)
+ Altitude (('1Byte signed'+109) * 25m (-127->-450m, 127->5900m))
                        3D Area (8):            Altitude bottom, top (each: ('1Byte signed'+109) * 25m (-127->-450m,
127->5900m), only once) +
                                                        n times: Position (1st absolute others compressed, see below)
                        3D Cylinder (9):        n times: Position (1st absolute others compressed, see below) + Radius
(1Byte in 50m, bit 7 scale 8x or 1x, bit 0-6) +
                                                        Altitude bottom, top (each: ('1Byte signed'+109) * 25m (-127-
>-450m, 127->5900m), only once)

------------

Remote Configuration (Type = 6)         NOTE: Do not use, in development!
Note: Signature (symmetric) is highly recommended. Skytraxx uses first 4byte of SHA1 + PSK
Note 2: Each reply feature with a suitable mask shall be played using round robin w/ 30sec intervals followed by a 3min
pause.
Note 3: Empty subtype removes the feature

[Byte 0]
bit 7-0         Subtype:
                        0:      Acknowledge configuration: Byte [1] subtype of ack
                        1:      Request. Byte[1] Subtype
                        2:      Position. Byte [1-6] latitude/longitude, Byte [7] altitude ('1Byte signed'+109) * 25m
(-127->-450m, 127->5900m), Byte [8] heading (encoded like in type 1)
                        3:      Reserved
                        4..8:   Geofence for Geo-Forwarding: Altitude bottom, top (each: ('1Byte signed'+109) * 25m
(-127->-450m, 127->5900m), only once) +
                                                        n times: Position (1st absolute others compressed, see below)
                        9..33:  Broadcast Reply feature. Byte[1] Wind Sectors (like in type 5), Byte [2] is type (and
forward bit) followed by its payload.
                                Recommendation: 9 for name. First 12 none-volatile, second 12 volatile

------------

Ground Tracking (Type = 7)
[recommended interval: floor((#neighbors/10 + 1) * 5s)]

[Byte 0-2]      Position        (Little Endian, 2-Complement)
bit 0-23        Latitude        (Absolute, see below)
[Byte 3-5]      Position        (Little Endian, 2-Complement)
bit 0-23        Longitude       (Absolute, see below)
[Byte 6]
bit 7-4         Type
                        0:    Other
                        1:    Walking
                        2:    Vehicle
                        3:    Bike
                        4:    Boot
                        8:    Need a ride
                        9:    Landed well
                        12:   Need technical support
                        13:   Need medical help
                        14:   Distress call
                        15:   Distress call automatically
                        Rest: TBD
bit 3-1         TBD
bit 0           Online Tracking

------------

HW Info (Type = 8)      (DEPRICATED)
[recommended intervall: very low, every 10min]

[Byte 0]        Instrument / Device Type (Manufacturer Spezific)
                Pull request 0x00 (has to be unicast, no further data)
                Manufacturer 0x01:                      (Skytraxx)
                        0x01 Wind station
                Manufacturer 0x06:                      (Burnair)
                        0x01 Base station Wifi
                Manufacturer 0x11:                      (FANET +)
                        0x01 Skytraxx 3.0
                        0x02 Syltraxx 2.1
                        0x03 Skytraxx Beacon
                        0x10 Naviter Oudie 5
```

```
                Manufacturer 0xFB:
                        0x01 Skytraxx WiFi base station
[Byte 1-2]      Firmware Build Date
bit 15                  0: Release 1: Develop/Experimental Mode
bit 9-14                Year from 2019 (0 -> 2019, 1 -> 2020, ...)
bit 5-8                 Month (1-12)
bit 0-4                 Day (1-31)
+ additional type/manufacturer/version spezific optional data
e.g. Recomendation / Skytraxx best practice:
Byte [3-4]
bit 15-4        Uptime in 30sec steps
bit 0-3         unused (Skytraxx Windstation: number used volatile replay features)

------------

Thermal (Type = 9)
[recommended intervall: floor((#neighbors/10 + 1) * 30s), if a thermal is detected]

[Byte 0-2]      Position of thermal     (Little Endian, 2-Complement)
bit 0-23        Latitude                (Absolute, see below)
[Byte 3-5]      Position of thermal     (Little Endian, 2-Complement)
bit 0-23        Longitude               (Absolute, see below)

[Byte 6-7]      Type                    (Little Endian)
bit 15          TBD, leave as 0
bit 14-12       confidence/quality      (0 = 0%, 7= 100%)
bit 11          Thermal Altitude Scaling 1->4x, 0->1x
bit 0-10        Thermal Altitude in m

[Byte 8]        Avg climb of thermal    (max +/- 31.5m/s, 2-Complement, climb of air NOT the paraglider)
bit 7           Scaling         1->5x, 0->1x
bit 0-6         Value           in 0.1m/s

[Byte 9]        Avg wind speed at thermal (max 317.5km/h)
bit 7           Scaling         1->5x, 0->1x
bit 0-6         Value           in 0.5km/h

[Byte 10]       Avg wind heading at thermal (attention: 90degree means the wind is coming from east and blowing towards
west)
bit 0-7         Value           in 360/256 deg

------------

HW Info (Type = A) (EXPERIMENTAL, will replace type 8)
[recommended intervall: very low, every 10min]

[Byte 0]        Header  (additional payload will be added in order 6 to 1, followed by Extended Header payload 7 to 0
once defined)
bit 7           Ping-Pong Request (must be unicast and must not cotain any data other then subheader, bits in header
are considered as requests)
bit 6           Hardware Subtype + Build Date
bit 5           ICAO address (+3byte address)
bit 4           Uptime (+2byte, time in minutes)
bit 3           Rx RSSI (+1byte RSSI+50, +3byte FANET address, example: -30 -> -80dBm, only valid for uni cast
requests, reply usually broiadcast)
bit 2-1         TBD
bit 0           Extended Header (+1byte directly after byte 0)

Hardware Subtype + Build Date
[Byte 1]        Instrument / Device Type (Manufacturer Spezific)
                Pull request 0x00 (has to be unicast, no further data)
                Manufacturer 0x01:                      (Skytraxx)
                        0x01 Wind station
                Manufacturer 0x06:                      (Burnair)
                        0x01 Base station Wifi
                Manufacturer 0x11:                      (FANET +)
                        0x01 Skytraxx 3.0
                        0x02 Syltraxx 2.1
                        0x03 Skytraxx Beacon
                        0x10 Naviter Oudie 5
                Manufacturer 0xFB:
                        0x01 Skytraxx WiFi base station
[Byte 2-3]      Firmware Build Date
bit 15                  0: Release 1: Develop/Experimental Mode
bit 9-14                Year from 2019 (0 -> 2019, 1 -> 2020, ...)
bit 5-8                 Month (1-12)
bit 0-4                 Day (1-31)


------------

Coordinate Formats

Compressed (reference coordinate required):
[Byte 0-1]      Position        (Little Endian, 2-Complement)
bit 0-15        Latitude
[Byte 2-3]      Position        (Little Endian, 2-Complement)
bit 0-15        Longitude

                        Details:
                        bit 15          even 0 odd 1 degree
                        ddeg = (signed 15bit) * value / 2^15
```

```
                        if(round(my_deg) is equal to bit15)
                                deg = round(my_deg) + ddeg
                        else
                                find minimum of |round(my_deg)-1 + ddeg - my_lat| and |round(my_lat1)+1 + ddeg -
my_lat|

                        (Max allowed distance 1deg -> approx. 111km latitude or longitude@latitude=equator,
                        longitude@latitude=60deg: 55km, longitude@latitude=70deg: 38km (critical))
                        (Max error <2m)
                        (Note: longitude block-bit could be extended by a further bit in case of lat > +/-60deg, future
work...)

Sample C code for decompressing:
float fns_buf2coord_compressed(uint16_t *buf, float mycoord)
{
        /* decode buffer */
        bool odd = !!((1<<15) & *buf);
        int16_t sub_deg_int = (*buf&0x7FFF) | (1<<14&*buf)<<1;
        const float sub_deg = sub_deg_int / 32767.0f;


        /* retrieve coordinate */
        float mycood_rounded = roundf(mycoord);
        bool mycoord_isodd = ((int)mycood_rounded) & 1;

        /* target outside our segment. estimate where it is in */
        if(mycoord_isodd != odd)
        {
                /* adjust deg segment */
                const float mysub_deg = mycoord - mycood_rounded;
                if(sub_deg > mysub_deg)
                        mycood_rounded--;
                else
                        mycood_rounded++;
        }

        return mycood_rounded + sub_deg;
}

Sample C code for compressing (note: byte order in stream is low to high):
uint16_t fns_coord2buf_compressed(float ref_deg)
{
        const float deg_round = roundf(ref_deg);
        const bool deg_odd = ((int)deg_round) & 1;
        const float decimal = ref_deg - deg_round;
        int dec_int = (int)(decimal*32767.0f);
        clamp(dec_int, -16383, 16383);

        return ((dec_int&0x7FFF) | (!!deg_odd<<15));
}


Absolute:
[Byte 0-2]      Position        (Little Endian, 2-Complement)
bit 0-23        Latitude
[Byte 3-5]      Position        (Little Endian, 2-Complement)
bit 0-23        Longitude

                        Details:
                        Latitude = value_lat/93206      \in [-90, +90]
                        Longitude = value_lon/46603 \in [-180, +180]
                        (Note: 32bit floating point is required for direct conversion)


------------

Signature (symmetric)

Use SHA1 and iterate over pseudo header (first 4 byte:  type + source address, were bits 6 and 7 of byte 0 are set to
0), over the payload, and over a pre-shared secret/key.
The first 4 byte of the resulting hash shall be interpreted as 32bit integer and put into the signature field (= normal
order due to little endian encoding).

------------

//todo address detection, etc...
//todo: as a base station is in rage, do not forward tracking info. only forward tracking info if very little traffic
is present...
//todo: forward bit for type 1 should only be set it no inet gateway in in close range

Notes:
------
Version number:
We omitted a bit field that shows the protocol version as this would take to much space. The app layer should provide
this, if required. (Todo)

Device ID:
-For unregistered Devices/Manufacturers: Set the Manufacturer to 0xFC or 0xFD and choose a random ID between 0x0001 and
0xFFFE.
List on the channel if the id is already used.
-0xFE shall be used for multicast (E.g. competition/group messaging).
-The manufacturers 0x00 and 0xFF as well as the IDs 0x0000 and 0xFFFF are reserved.
```

Manufacturer IDs:

0x00	[reserved]
0x01	Skytraxx
0x03	BitBroker.eu
0x04	AirWhere
0x05	Windline
0x06	Burnair.ch
0x07	SoftRF
...	
0x11	FANET+ (incl FLARM. Currently Skytraxx, and Naviter)
...	
0xE0	OGN Tracker
...	
0xFB	Espressif based base stations, address is last 2bytes of MAC
0xFC	Unregistered Devices
0xFD	Unregistered Devices
0xFE	[Multicast]
0xFF	[reserved]

Reserved for compatibility issues:

0xDD
0xDE
0xDF
0xF0
0x20