

### 3 types de message :

- 1 - Position =  $x, y, z$
- 2 - GPS = numSat, dop
- 3 - Status = error, 'softwar-version'

Emetteur: choix par le type de message

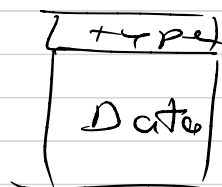
type énuméré : eTypeMsg

E-POSITION	1
E-GPS	2
E-STATUS	3

Structures : sPosition      3 double  $x, y, z$   
                   : sGPS            uint32\_t, double  
                   : sStatus        bool, 'chr [5]

msg :

- Type      type énuméré
- Data      1 des 3 Structures.

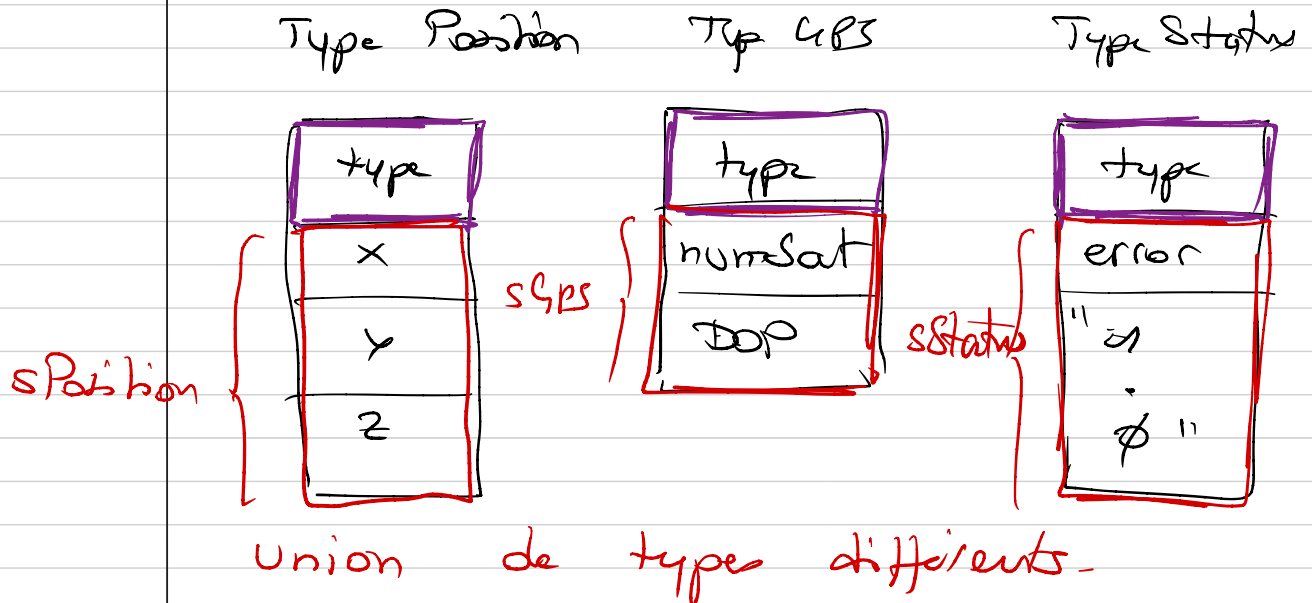


le contenu de Data  
 varie en fonction de type

msg: variable dont le contenu (structure)  
 se changeant en fonction du type.

Si type = E-POSITION → Data doit être interprété  
 comme une sPosition  
 E-GPS                      sGPS  
 E-STATUS                sSTATUS

Variable msg:



typedef union {

sPosition pos;

sGPS gps;

sStatus status;

} uData;

uData d;

d.pos	• x • y • z	d.gps.numSat • dop	d.status = error • SW Version
-------	-------------------	-----------------------	----------------------------------

```

typedef union {
    sPosition pos;
    sGPS gps;
    sStatus status;
} uData;

```

```

typedef struct {
    eTypeMsg type;
    uData data;
} sMsg;

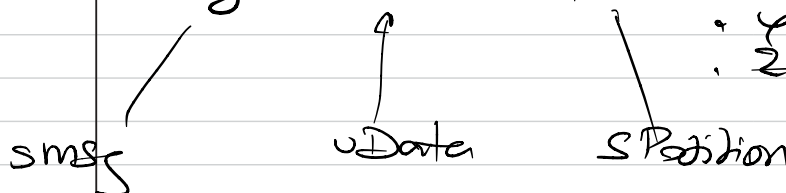
```

1	type
1	x
2	y
3	z

```

sMsg msg;
msg.type = E_Position;
msg.data.pos.x = 1;
           y = 2;
           z = 3;

```



TD 20200721

• h → eTypeMsg  
 3 struct sPosition/sGPS/sStatus  
 1 union uData  
 1 struct sMsg