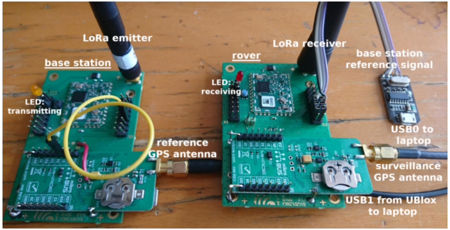
Long range wireless communication for centimeter-resolution real-time-kinematic GNSS positioning: locating sensors on an Arctic glacier

Jean-Michel Friedt[1], Éric Bernard[2], Florian Tolle[2], Didier Donsez[3]

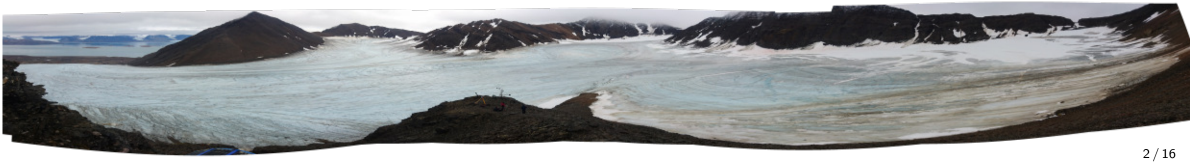
[1] FEMTO-ST/Time & Frequency, UMR 6174, Besançon, France
[2] ThéMA, UMR 6049, Besançon, France
[3] LIG/ERODS, UMR 5217, Grenoble, France



January 19, 2023

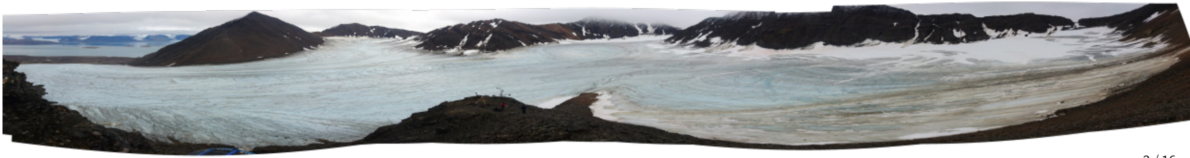
Context: Austre Lovén glacier (79°N, Spitsbergen)

- Long term monitoring (2007–today) of an Arctic glacier and its hydrological budget ...
- ... using various classical (ablation stakes) and dedicated instrumentation.
- Use of cm-resolution GNSS for
  - Digital Elevation Model mapping (v.s. LiDAR)
  - glacier motion monitoring
  - recovering sensors buried under snow (April)



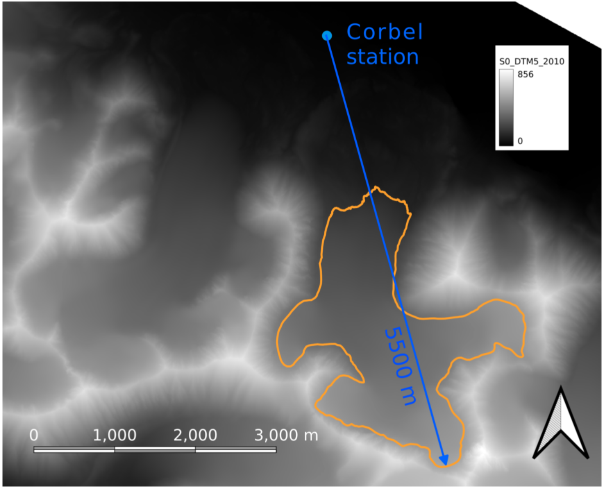
Context: Austre Lovén glacier (79°N, Spitsbergen)

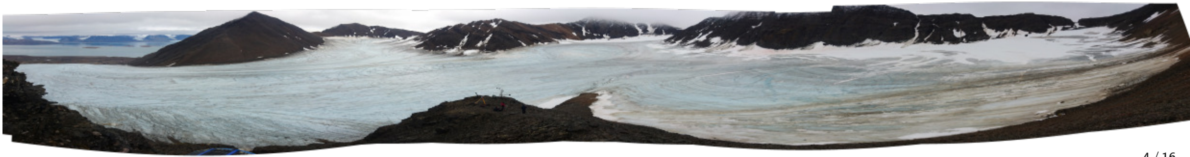
- Long term monitoring (2007–today) of an Arctic glacier and its hydrological budget ...
- ... using various classical (ablation stakes) and dedicated instrumentation.
- Use of cm-resolution GNSS for
  - Digital Elevation Model mapping (v.s. LiDAR)
  - glacier motion monitoring
  - recovering sensors buried under snow (April)



Context: Austre Lovén glacier (79°N, Spitsbergen)

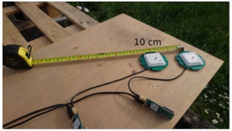
- Long term monitoring (2007–today) of an Arctic glacier and its hydrological budget ...
- ... using various classical (ablation stakes) and dedicated instrumentation.
- Use of cm-resolution GNSS for
  - Digital Elevation Model mapping (v.s. LiDAR)
  - glacier motion monitoring
  - recovering sensors buried under snow (April)

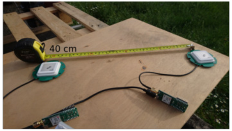


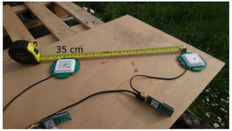


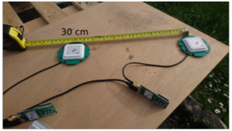
Zed-F9P & RTKLib [1]

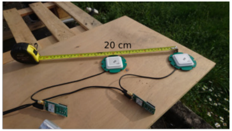
- Availability of affordable multiconstellation, multiband GNSS receiver with phase information (UBlox Zed-F9P)
- Availability of an **opensource** differential GNSS analysis library: RTKLib and its extension Demo5
- Need to make the (mobile) rover communicate with the (static) basestation to compensate for common sources of error (ionospheric, tropospheric delays: $\pm$7 ns=$\pm$2.1 m)
- Preliminary demonstration using a short baseline with rover & basestation wired to computer

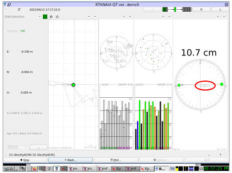


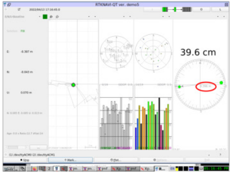


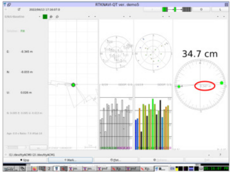


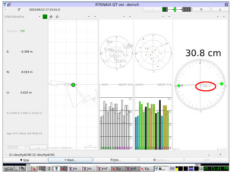


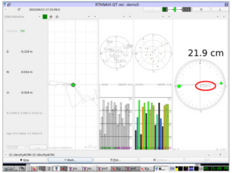












[1] https://github.com/rinex20/RTKLIB-demo5: "A version of RTKLIB optimized for single frequency low cost GPS receivers, especially u-blox receivers."

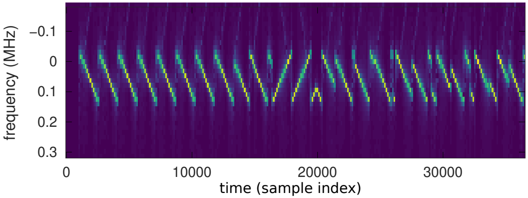
Centipede? (https://docs.centipede.fr/)

- Relies on a mobile phone infrastructure to broadcast basestation reference signals
- Ny-Ålesund: hosts multiple radiotelepscopes and low earth orbiting communication stations
  $\Rightarrow$ no radiofrequency emission above 1 GHz

⟶

LoRa-based point to point communication

- LoRa ("Long Range"): proprietary radiofrequency communication protocol developed by Semtech
- Undocumented cryptographic layer but up/down **chirp** with known bandwidth and duration $\Rightarrow$ ability to compute link budget and signal to noise ratio improvement from pulse compression
- sub-GHz (868 MHz), low datarate but long range: compatible with UBX format communication from basestation to rovers?
- datarate v.s sentence length in the UBX protocol [2]: RAWX sentences $16 + 32 \times N$ bytes with N number of satellites
- All constellation and all channels: $\sum$ GPS (16), Galileo (18), Beidou (5), GLONASS (12), SBAS (3) & QZSS (4)
  =58 channels=1872 bytes/s=14976 bits/s
- only $B = 500$ kHz & $SF = 7$ can reach $C = 21875$ b/s $\Rightarrow$ range loss



datarate C (bits/s) depending on chirp duration 2^{SF} and bandwidth B

$$C = \frac{SF}{2^{SF}} \times \frac{4}{5} B$$

at *Coding Rate* $CR = 1$

$$\text{Friis (energy conservation): } \frac{P_R}{P_E} = \frac{\lambda^2}{(4\pi)^2 d^2}$$

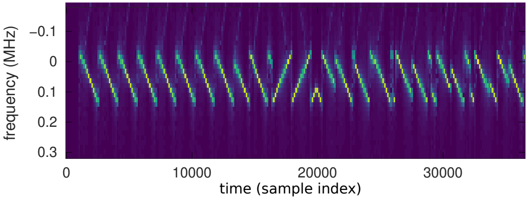
$$\Rightarrow FSPL = 20\log_{10}(d) + 20\log_{10}(f) - 147.55$$

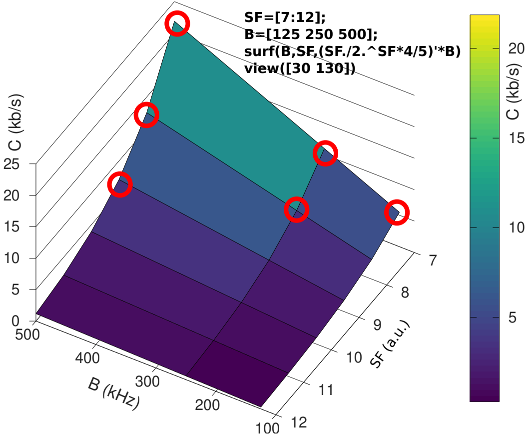
Detection limit given by $N = k_B \cdot T \cdot B$ but signal to noise ratio improvement from pulse compression $PCR = SF \times B \Rightarrow SF \times 2 = PCR \times 2 = range \times \sqrt{2}$

[2] `https://cdn.sparkfun.com/assets/f/7/4/3/5/PM-15136.pdf` p. 182)

LoRa-based point to point communication

- LoRa ("Long Range"): proprietary radiofrequency communication protocol developed by Semtech
- Undocumented cryptographic layer but up/down **chirp** with known bandwidth and duration ⇒ ability to compute link budget and signal to noise ratio improvement from pulse compression
- sub-GHz (868 MHz), low datarate but long range: compatible with UBX format communication from basestation to rovers?
- 1872 bytes=18720 bits transfered using RS232/8N1 last 162.5 ms @ 115200 bauds and resulting 14976 bits require 685 ms
- reduce number of channels by limiting to GPS only in order to increase SF ⇒ increase range
- One-way point-to-point: use only **physical layer** LoRa and *not* routing LoRaWAN





RIOT-OS (2009–) implementation

Yet another executive environment (monolithic binary output without dynamic library or dynamic executable application loading) [2] with

- high-abstraction including Hardware Abstraction Layer, drivers, multithreading, scheduler, mutex...
- aims at supporting multiple hardware platforms (STM32 based im880b & LoRa-E5B)
- provides a complete set of *test* cases.

```
gpio_t jmf_gpio_in= GPIO_PIN(0,12); // PA12 = P6 - USART1-RTS
static sx127x_t sx127x;

int main()
{gpio_init(jmf_gpio_in, GPIO_IN_PU);
 sx127x.params = sx127x_params[0];
 netdev_t *netdev = &sx127x.netdev;
 netdev->driver = &sx127x_driver;
 netdev->driver->init(netdev);
 netdev->event_callback = _event_cb;
// 250 kHz, SF=8 => 6.25 kbps CR=1..4 overhead 1.25, 1.5, 1.75, 2
 lora_setup_cmd(250,8,5);
 channel_cmd(868000000);  // CR fixed at 4/5 (CR=1) for LoRaWAN
...
}
```

Two threads, one in charge of data collection and one in charge of transmitting, switching under timer timeout condition (end of transmission)

- basestation: collect RS232 sentences and transmit over LoRa
- rover: collect LoRa sentences and transmit over RS232

[2] after RTEMS (1993–), eCos (1998–), TinyOS† (2000–2012), FreeRTOS (2003–), NuttX (2007–), ChibiOS (2007–)...

RIOT-OS: rover or basestation at compile time (pre-processor), timing constrainted FSM

```
void *_rstx_thread(void *arg)
{ (void)arg;
  char val;
  static msg_t _msg_q[SX127X_LORA_MSG_QUEUE];
  msg_init_queue(_msg_q, SX127X_LORA_MSG_QUEUE);
  while (1)
    {stdio_read (&val,1);   // USEMODULE += shell in Makefile
     if (compteurin<BUFSIZE)
       {message[compteurin]=val;
        compteurin++;
}    }  }

int main()
{...
#ifndef RXROVER
 _rstx_pid=thread_create(stacktx,sizeof(stacktx),THREAD_PRIORITY_MAIN-1,
   THREAD_CREATE_STACKTEST, _rstx_thread, NULL, "rstx_thread");
 oldcompteurin=0;
 while (1)
  {last_wakeup = xtimer_now();
   xtimer_periodic_wakeup(&last_wakeup, interval);
   if (compteurin>0)                  // RS232 received
    {if (compteurin==oldcompteurin)  // but long silence since
      {indice=0;
       wdt_kick();
       do{if (compteurin>=SX127X_LORA_MSG_QUEUE)
          {send_cmd(&message[indice],SX127X_LORA_MSG_QUEUE);
           indice+=SX127X_LORA_MSG_QUEUE;
           compteurin-=SX127X_LORA_MSG_QUEUE;
           last_wakeup = xtimer_now();
           xtimer_periodic_wakeup(&last_wakeup, 30);
          }
          else
          {send_cmd(&message[indice],compteurin);
           compteurin=0; oldcompteurin=0;
          }
      }  } while (compteurin>0);
     else {oldcompteurin=compteurin;}
  } }
...
```

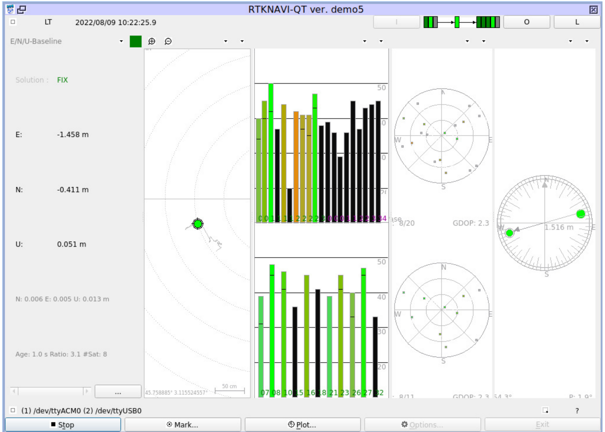
← Basestation ↓ Rover

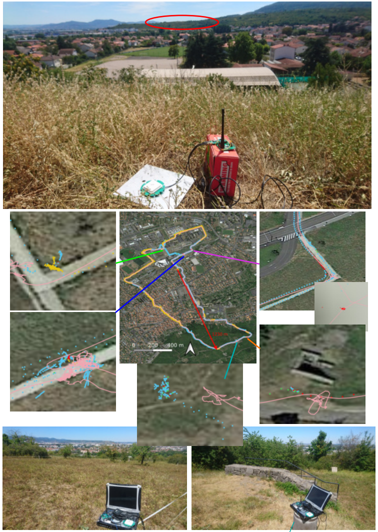
```
void *_rsrx_thread(void *arg)
{(void)arg;
 static msg_t _msg_q[SX127X_LORA_MSG_QUEUE*2];
 msg_init_queue(_msg_q, SX127X_LORA_MSG_QUEUE*2);
 xtimer_ticks32_t last_wakeup;
 uint32_t interval = 25600*2;
 unsigned int oldindexin=0;

 while (1)
  {last_wakeup = xtimer_now();
   xtimer_periodic_wakeup(&last_wakeup, interval);
   if (indexin>0)
    {if (oldindexin==indexin)
      {if (indexin>BUFSIZE) {indexin=BUFSIZE;} // too many items
       wdt_kick();
       indexin=0; // content of buffer has been dumped, restart
      }
      else {oldindexin=indexin;} // still receiving
    }
  }
}

int main()
{...
#ifdef RXROVER
 _rsrx_pid =  thread_create(stackrx, sizeof(stackrx), →
       ↪THREAD_PRIORITY_MAIN - 1,
   THREAD_CREATE_STACKTEST, _rsrx_thread, NULL,
   "rsrx_thread");
 indexin=0;
 listen_cmd(); // infinite loop
...
}
```

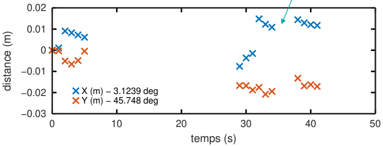
Preliminary results (Clermont-Ferrand)

Single	=no LoRa communication
Float	=searching for solution
Fix	=cm RTK solution

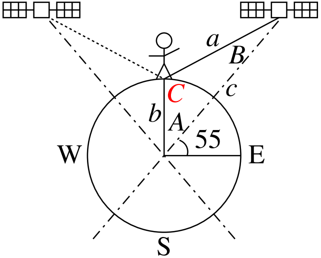
⇒ real time feedback on solution evolution, 1.5 km line of sight







Results: Spitsbergen (high latitude ⇒ poor coverage of GPS/Galileo)



GPS inclination: 55°

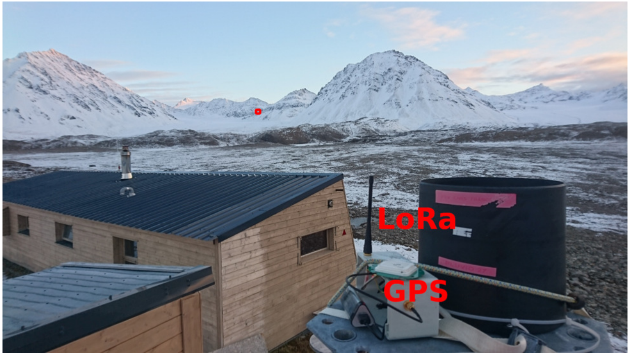
$A = latitude - 55 \Rightarrow C$ elevation of satellite from viewer?

$b = 6400$ km and $c = 6400 + 20200$ km are known

$\Rightarrow a^2 = b^2 + c^2 - 2bc\cos(A) = 21671$ km if observer at North Pole

$\Rightarrow \cos(C) = \frac{a^2+b^2-c^2}{2ab} = 45°$.

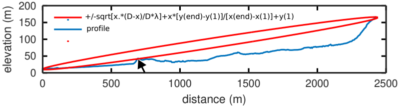
Spitsbergen: $C \leq 59°$ towards South but 32° towards North

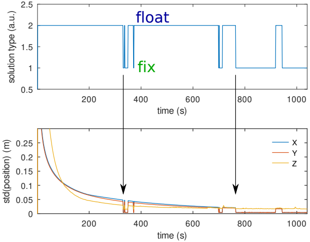


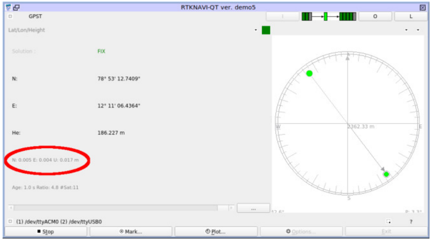
Results: Spitsbergen (high latitude ⇒ poor coverage of GPS/Galileo)

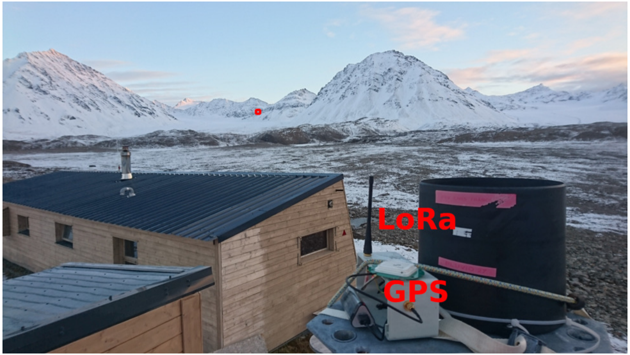
Range: $D = 2.4$ km (moraine)
Accuracy: < 2 cm
Convergence duration ≤ 15 min
Fresnel zone clearance:

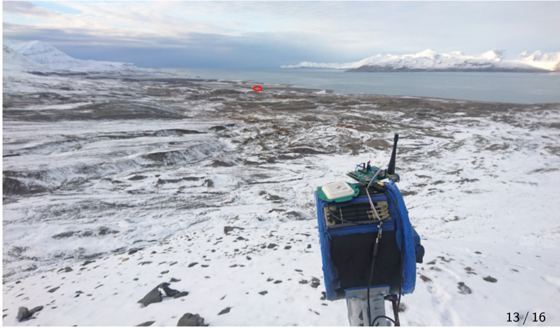
$$R > \frac{1}{2}\sqrt{\frac{cD}{f}} \simeq 15 \text{ m}$$







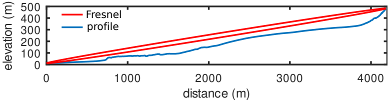


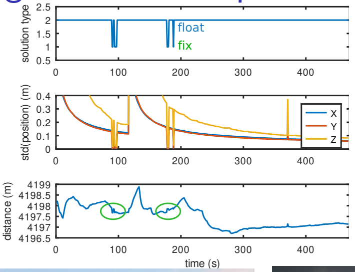


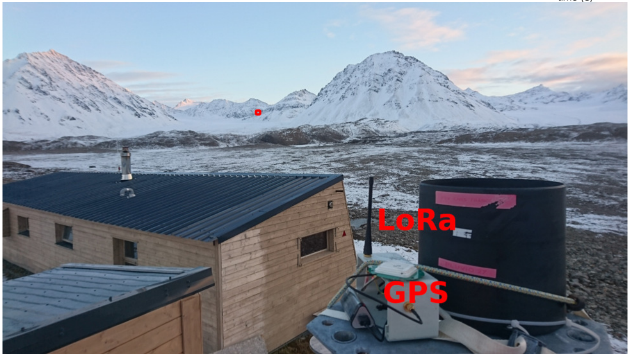
Results: Spitsbergen (high latitude ⇒ poor coverage of GPS/Galileo)

Range: 4.2 km (ridge)
Accuracy: < 2 cm
Convergence duration ≤ 8 min

Fresnel zone analysis:

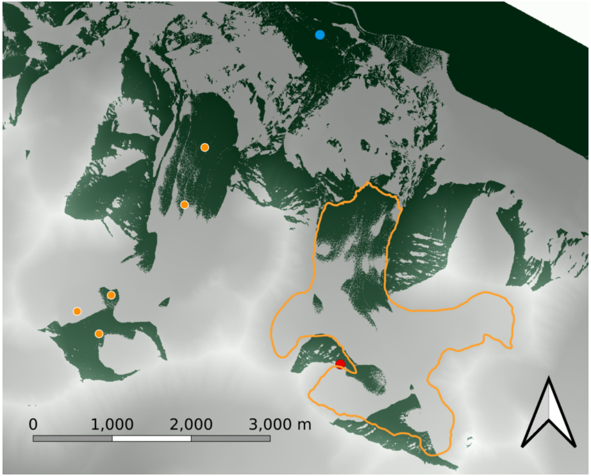


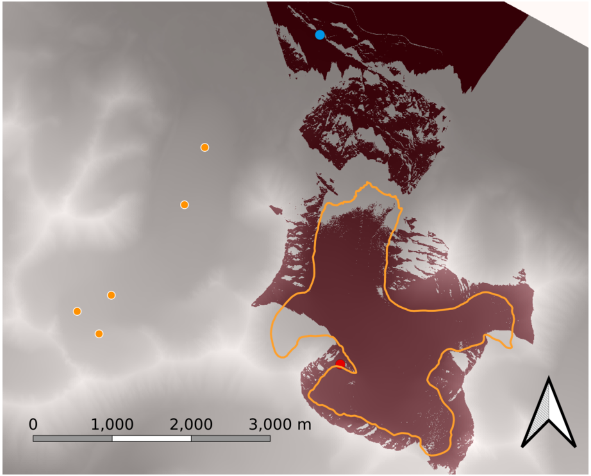




Radiofrequency communication range (QGis: viewshed analysis)

Using Digital Elevation Model of the area being surveyed to predict communication range





From Corbel station (blue dot)
including Midtre Lovén glacier (S. Filhol, Oslo Univ.)

From the ridge (red dot)
blue+red = full coverage

Conclusion & work in progress

- Functional demonstration of broadcasting basestation messages to rovers
- Functional cm-resolution measurement up to 6-km range (expected 20 km line of sight from SNR)

Work in progress:

- **Multihop**: topography constrained makes endpoint-gateway communication impractical
- Attempt at using an endpoint as **relay**: restricts communication to a **single** frequency/SF
- Standard being published but not (yet?) implemented.
- Selecting a better suited exchange **protocol** than UBX format (RTCM? Compact SSR? SPARTN?)
- Get RIOT-OS to work! (SPI test broken!)

Github repository:
`https://github.com/jmfriedt/RIOT_NyAlesund`

Reference:
J.-M Friedt, *Communication LoRa au moyen de RIOT-OS pour la mesure centimétrique par GPS différentiel avec RTKLib*, Hackable **45** (Nov-Dec. 2022)