LPS implementation

Goal

An understanding of the hardware design, where the algorithms are implemented and where to find the code. Enough knowledge to modify the current implementation.

Parts

- Node (Anchor and sniffer)
- LPS Deck for Crazyflie
- Roadrunner

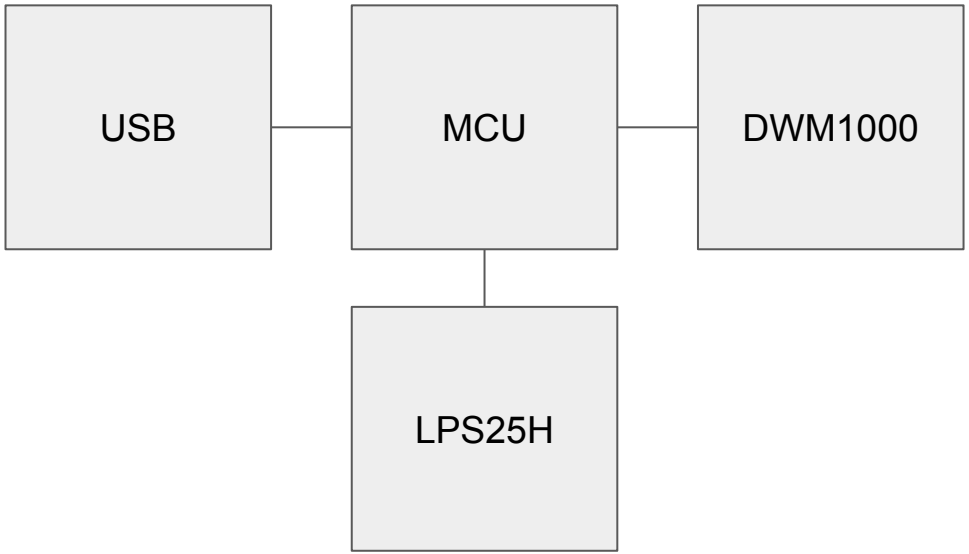
libdw1000

- Library for accessing dw1000 chip
- Platform independent
- Used in both Node and Crazyflie
- https://github.com/bitcraze/libdw1000

Node hardware

- STM32F072 MCU (Cortex-M0, 48MHz, 16kb SRAM, 128kb flash)
- DWM1000
- High precision pressure sensor (LPS25H)
- USB

Node architecture



Node firmware

- FreeRTOS
- Libdw1000
- Simplistic, one file per mode (TWR, TDoA2, TDoA3)
- Configuration
  - DW1000
  - ID (address)
  - Position
- https://github.com/bitcraze/lps-node-firmware
  - TWR in src/uwb_twr_anchor.c
  - TDoA2 in src/uwb_tdoa_anchor2.c
  - TDoA3 in src/uwb_tdoa_anchor3.c

Ranging error management strategy

- Sequence numbering of packets
- Clock correction noise threshold
- Clock correction low pass filter
  - Reset in case of many outliers

TDoA3 memory context

- Id
- Sequence number
- TX time
- TX randomization state
- List of remote anchor data
- List of remote ids to include in data section of TX messages
- Set of anchors seen

TDoA3 memory context - remote anchor data

- Sequence number
- RX timestamp
- TX timestamp
- Distance
- Distance update time
- Clock correction
- Data validity status

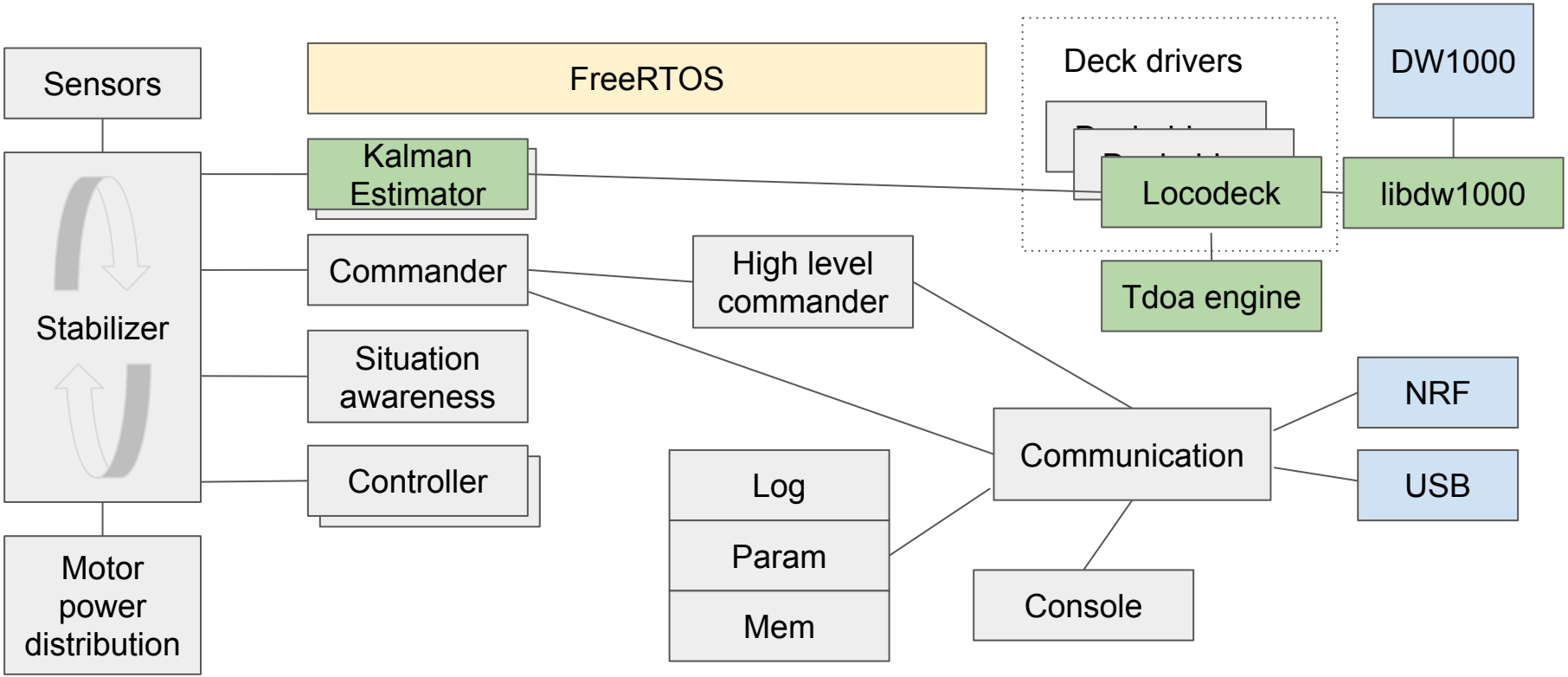
TDoA3 memory management strategy

- Limited memory space
- 16 anchor slots
  - Pruned at regular intervals (1s)
  - A random set of 16 is chosen out of all available anchors

LPS deck

- Used on the Crazyflie
- Integrated in the Roadrunner
- DWM1000 chip
- SPI

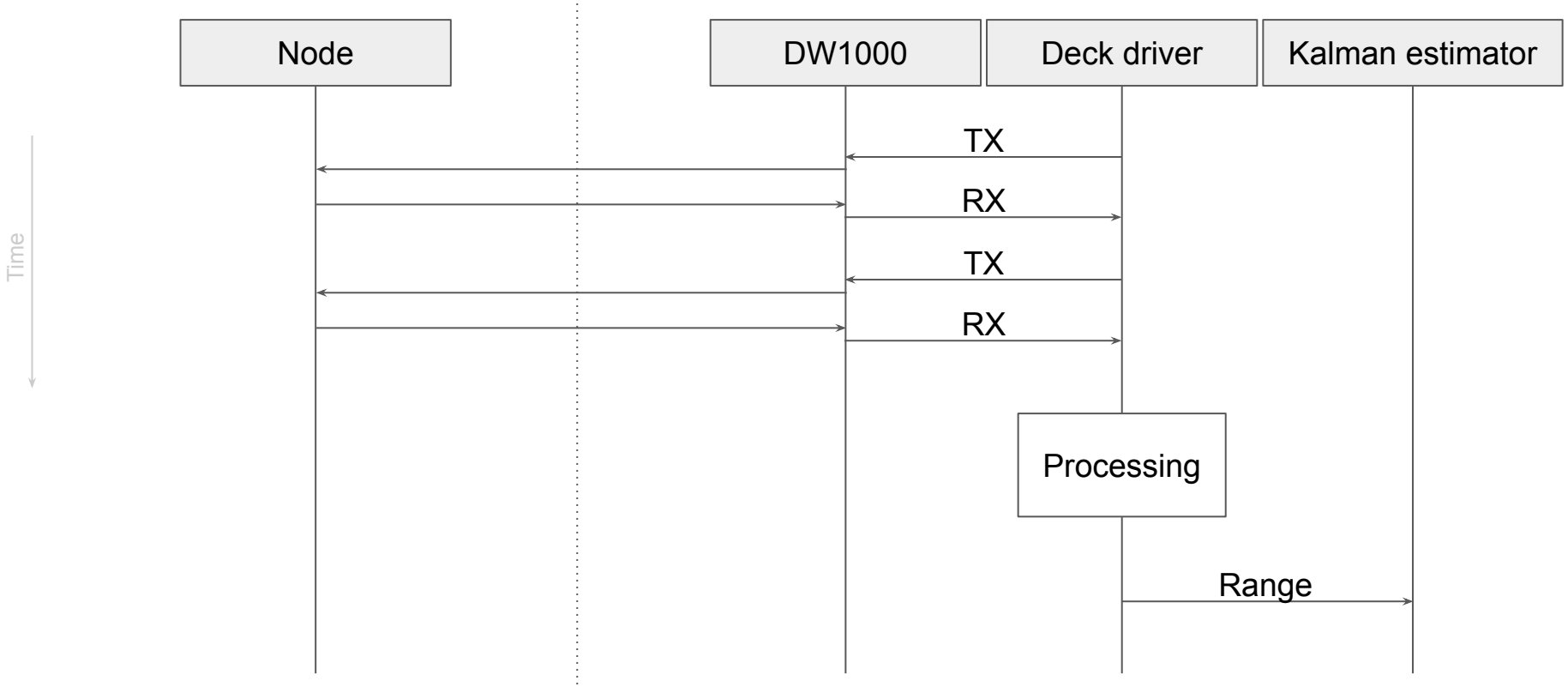
Crazyflie firmware overview with LPS deck



Crazyflie firmware LPS deck driver

- Runs in separate task
- TWR, TDoA2, TDoA3
- Event model
- Uses the kalman estimator (automatically enabled)
- Feeds range and tdoa measurements into the kalman filter for position estimation
- Memory mapping for anchor positions
- Log variables for various properties
- Reduced transmission power of radio in CF
- src/deck/drivers/src

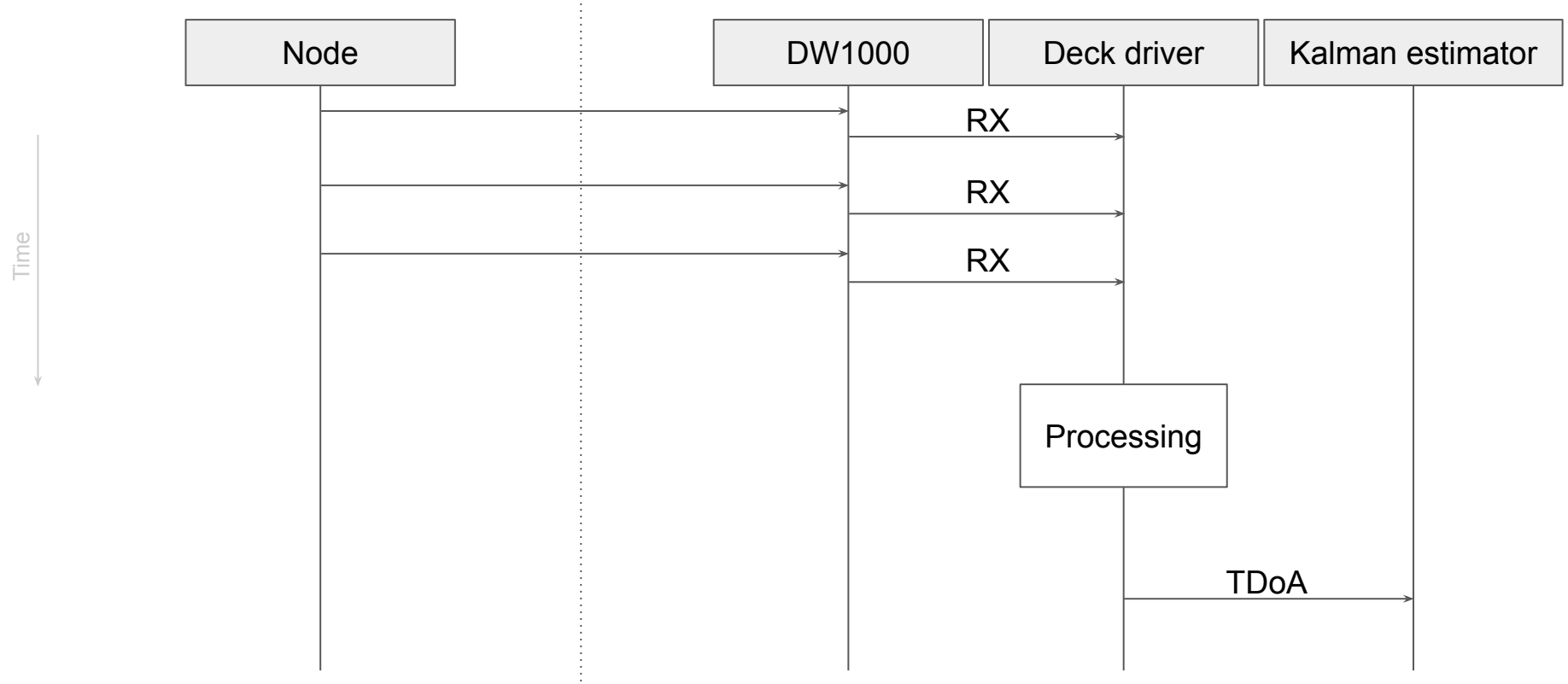
Crazyflie firmware, LPS flow chart, TWR



Two Way Ranging implementation

- Implemented in src/deck/drivers/src/lpsTwrTag.c

Crazyflie firmware, LPS flow chart, TDoA



TDoA implementation

- TDoA2 in src/deck/drivers/src/lpsTdoa2Tag.c
- TDoA3 in src/deck/drivers/src/lpsTdoa3Tag.c
  - Mainly data management
  - Uses the TDoA engine for calculations and storage

TDoA engine

- Generalized TDoA implementation
- Clock correction
- Anchor matching
  - Semi-randomized process
- TDoA calculation
- Separate data storage for dynamic handling of anchors
- TDoA engine and storage code in src/utils/src/tdoa

TDoA storage memory structure

- List of anchor contexts (16)
  - Id
  - TX time
  - RX time
  - Sequence number
  - Clock correction data
  - Anchor position
  - List of Time of Flight (16), reported by the anchor
  - List of remote anchor data (16), reported by the anchor

TDoA - ToF and Remote anchor data

- Time of Flight data
  - Id
  - ToF
- Remote anchor data
  - Id
  - Sequence number
  - RX time

TDoA storage memory management strategy

- Store as much as possible of the data from the anchors in static sized lists
- Data is timestamped
- Replace the oldest instance

TDoA ranging error strategy

- Sequence numbering of packets
- Clock correction noise threshold
- Clock correction low pass filter
  - Reset in case of many outliers
- Diffuse anchor-anchor ranging errors by randomizing TDoA anchor pairs
- TDoA outlier filter

TDoA outlier filter

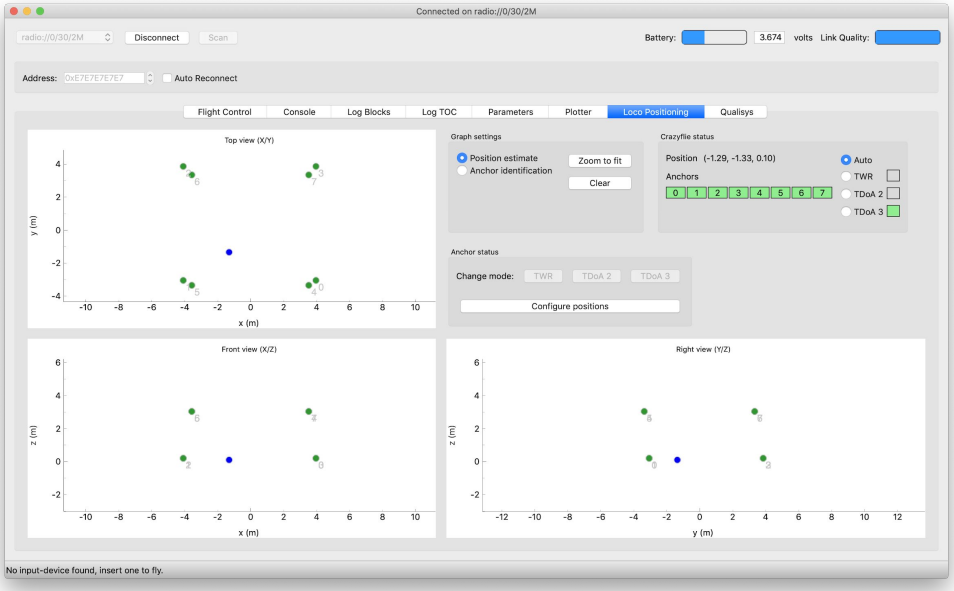
- Compare TDoA sample to current estimated position
- Reject if too far away
  - Dynamic acceptance range
  - Leaky bucket based
- Accept all samples when too many outliers

Python lib LPS support

- Memory mapping for anchor positions
- Helpers for sending data from Crazyflie to LPS Nodes (via UWB LPP packet)
  - Mode change
  - Setting anchor position
- https://github.com/bitcraze/crazyflie-lib-python/tree/0.1.8/lpslib

LPS support in python client

- Anchor status
- Anchor position configuration
- Mode change
- Code in
  src/cfclient/ui/tabs/locopositioning_tab.py



Conclusions

- Hardware: Node and deck
- Firmware in Node and Crazyflie firmware
- Supporting lib/client in python