Baseband Signal Processing Framework for the OsmocomBB GSM Protocol Stack

Harald Kröll, Christian Benkeser, Stefan Zwicky,
Benjamin Weber, Qiuting Huang

Integrated Systems Laboratory, ETH Zurich

June 27, 2012

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Outline

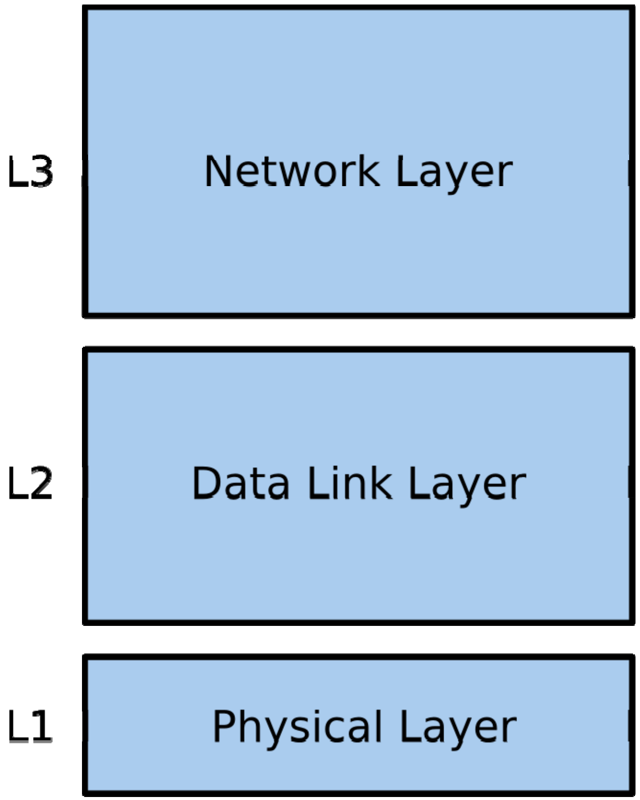
- Introduction into GSM and OsmocomBB
- Framework and interface
- Testbed architecture and setup
- Conclusion

GSM and Open Source

- Facts
  - Most ubiquitous cellular standard
  - 5 billion subscribers (2010)
  - Phones on the market since 1992
  - Very few baseband vendors
- Open Source in GSM
  - OpenBTS (since 2007/08)
  - AirProbe (since 2007)
  - OpenBSC (since 2008)
  - OsmocomBB (since 2010)
  - …

GSM and OsmocomBB

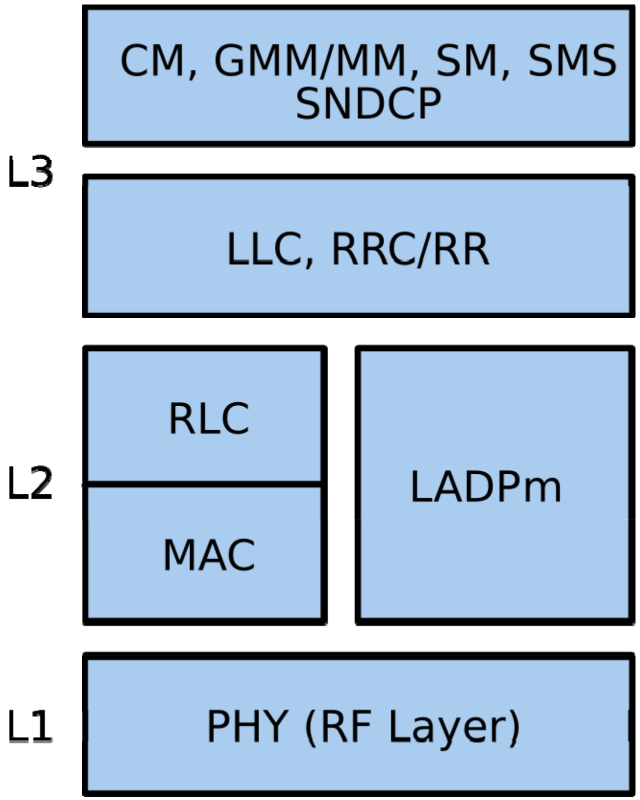
- GSM Protocol Layers, simplified overview



  - Relationship to OSI protocol layers
  - Influences from various specifications (GERAN/UTRAN)

GSM and OsmocomBB

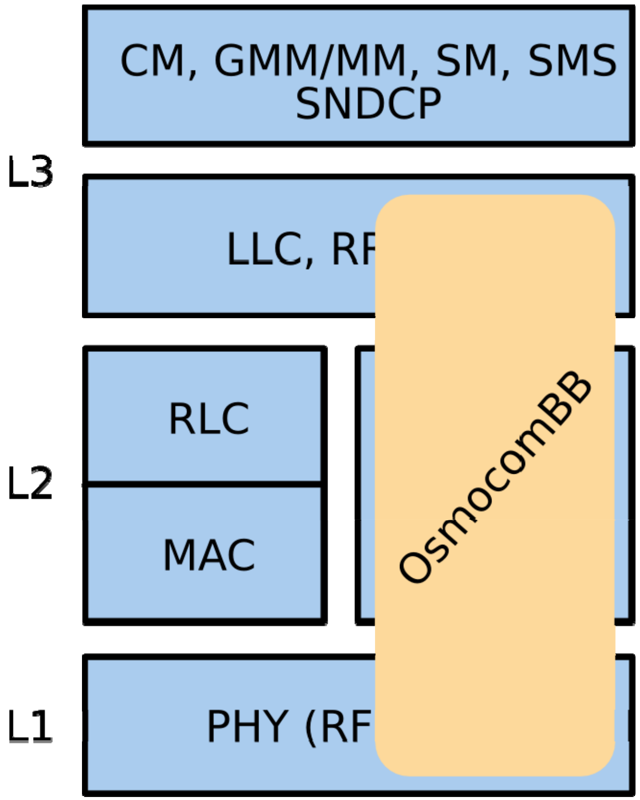
- GSM Layers, various protocols



  - Influences from
    - *A/Gb* (pre release 5 terminals)
    - *Iu* (release 5 terminals, UMTS interface)

GSM and OsmocomBB

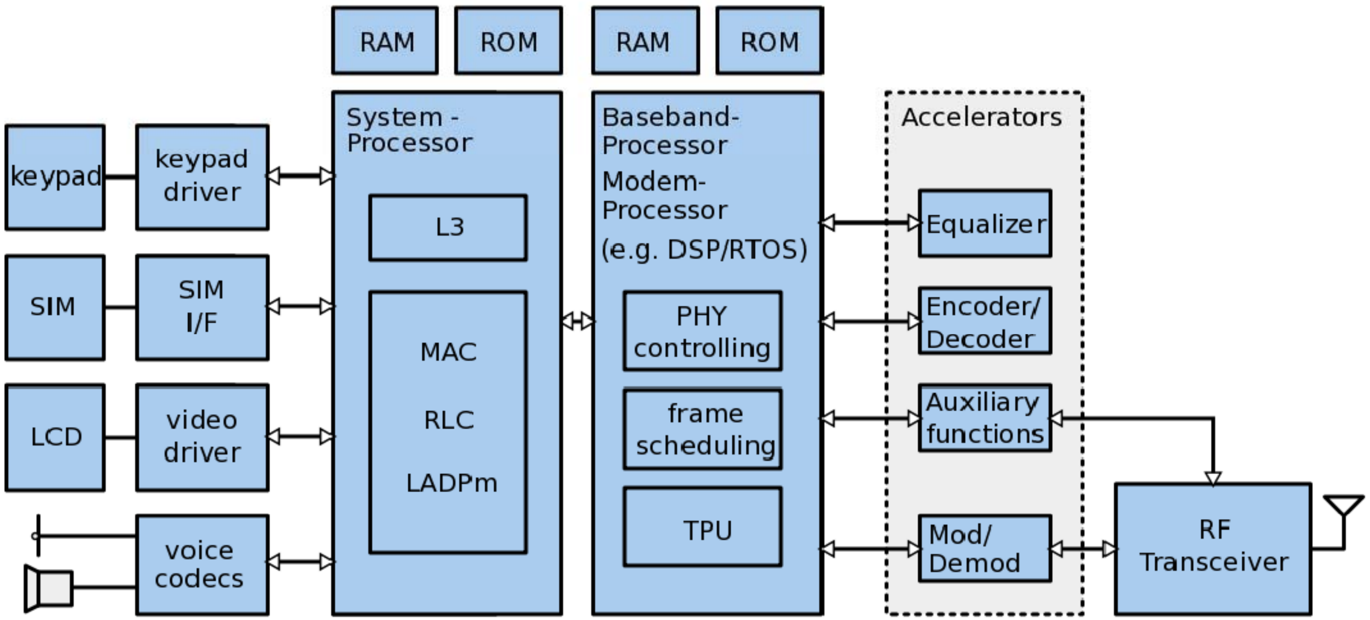
- OsmocomBB



- Open Source GSM Baseband software
- Implementation of L2/L3 in C running on a host PC
- Low cost *feature phones* used as L1
- “Limited” PHY support
- Interfacing of baseband processors (e.g. TI’s Calypso)

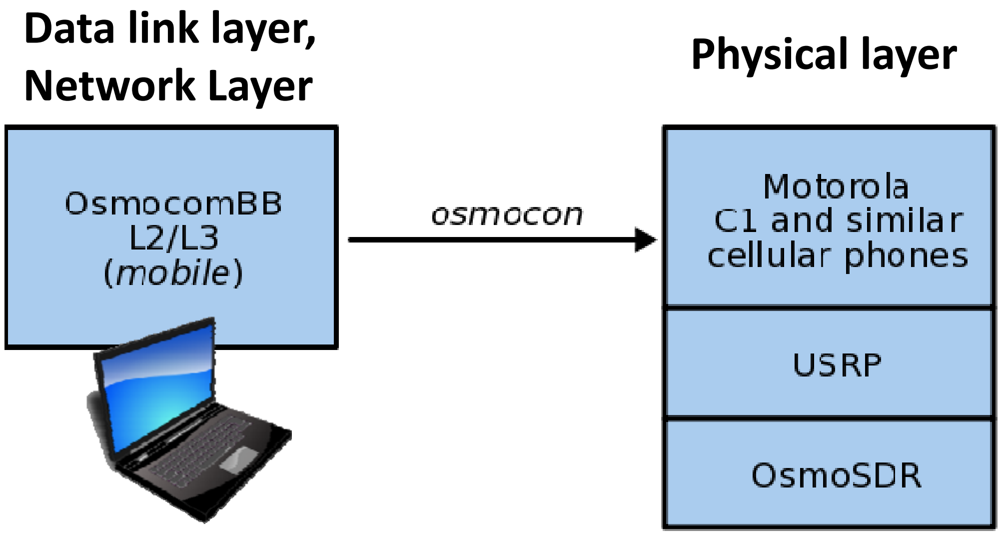
Architecture of a Feature Phone

- Baseband processor / modem processor (Qualcomm pat.)
- Computationally intensive tasks in accelerator blocks



OsmocomBB Setup

- Baseband and protocol stack

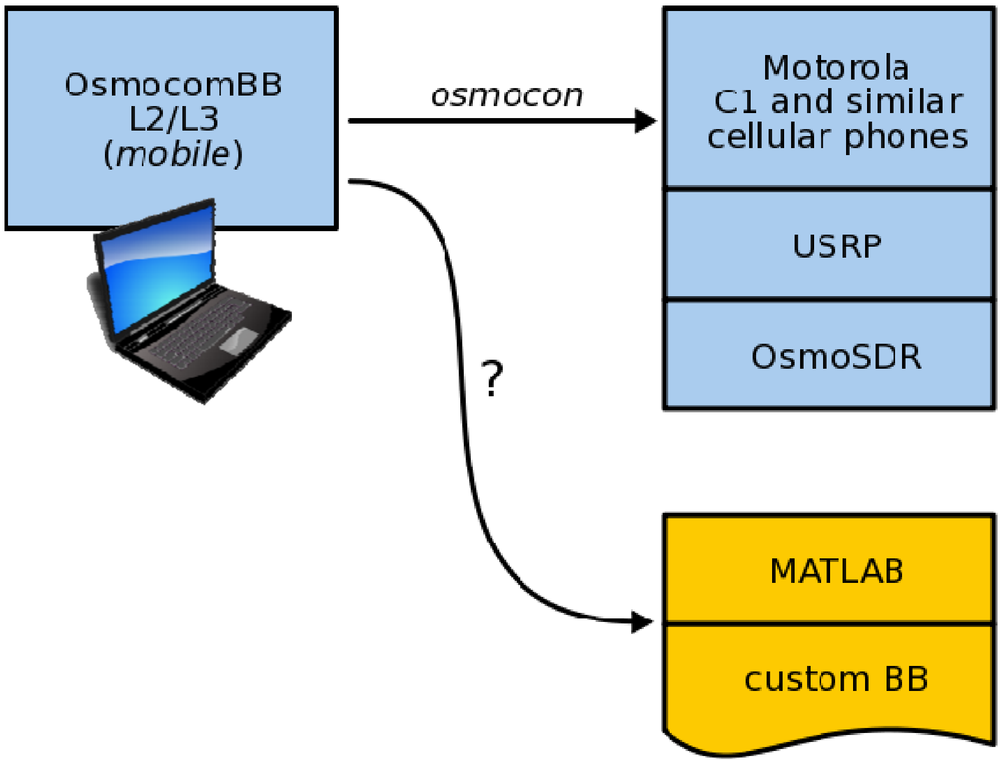


OsmocomBB Setup

- Baseband and protocol stack

**Data link layer,
Network Layer**

Physical layer



Support of an ubiquitous scientific computer language like GNU/Octave or Matlab is missing

Prospects of running a complete GSM stack

- New approaches during PHY development
  - Simulation of PHY together with L2/L3
  - Interaction between PHY and higher layers
- PHY development: controlling, debugging, visualization
  - Reporting of measurement data to display of phone
  - En-/disabling specific PHY functions from user interface
- Hybrid ARQ schemes, incremental redundancy (IR)
  - Interaction between channel decoding and MAC layer
  - Improved average throughput evaluation
- **A flexible interface between L1 and upper layers allows crossing layer boundaries**

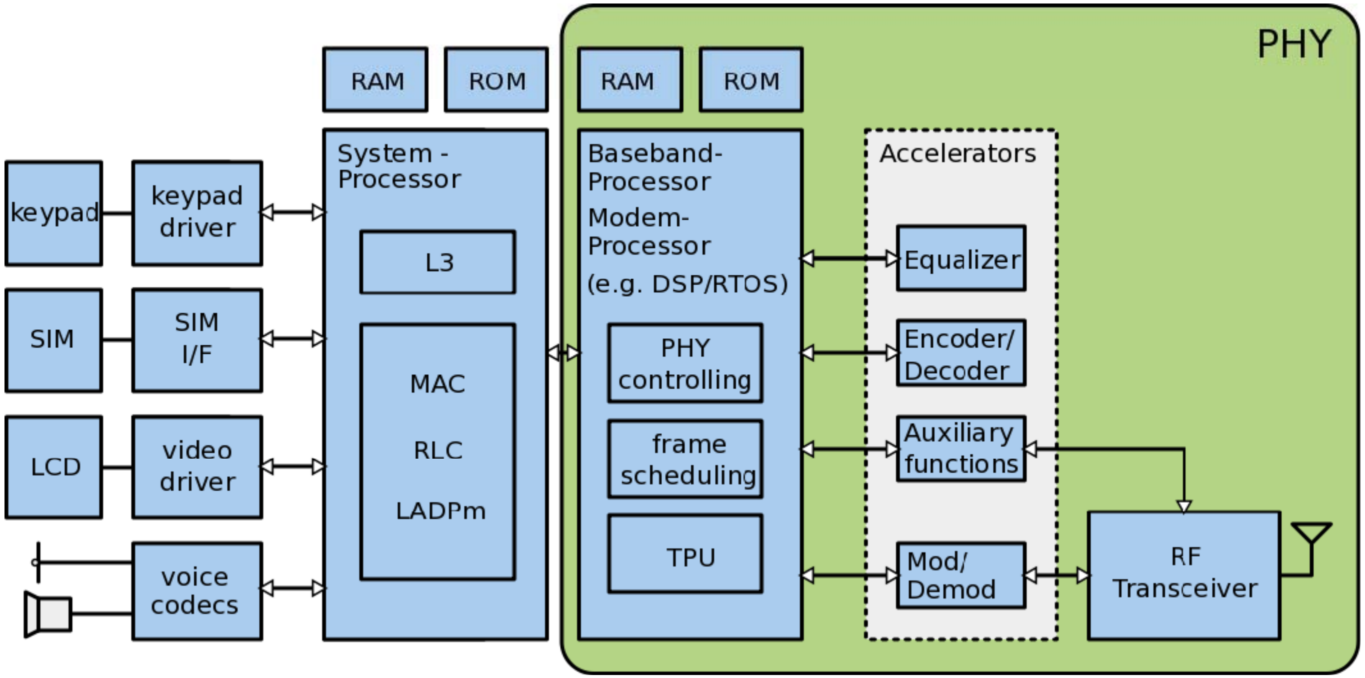
Interface between L1 and L2

- No standardized interface specified
- 3GPP foresees *primitive messages*
  - Request (REQ), confirm (CONF) and indication (IND)
- L1CTL from OsmocomBB
  - Message examples

Functionality	**L1CTL messages**
Reset PHY	L1CTL_RESET_REQ
	L1CTL_RESET_CONF
Power Measurement	L1CTL_PM_REQ
	L1CTL_PM_CONF
Synchronization	L1CTL_FBSB_REQ
	L1CTL_FBSB_CONF

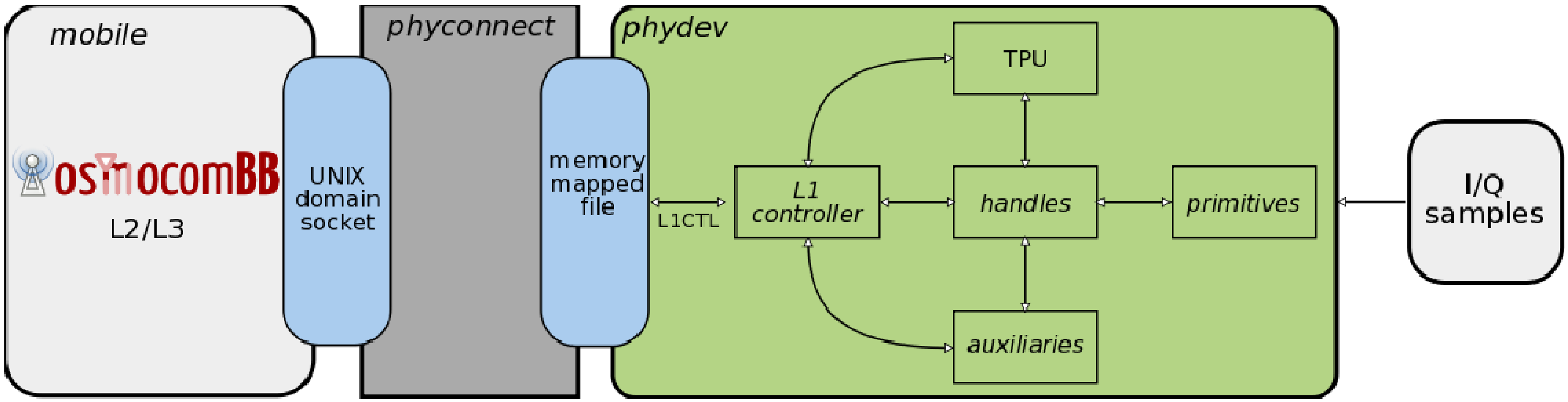
Proposed Signal Processing Framework

- Goals
  - Map complete PHY to Matlab
  - L1CTL interface to simplify operation with OsmocomBB



Framework Overview

- *mobile:* OsmocomBB application running L2/L3
- *phyconnect*: Interface to connect *mobile* to Matlab via unix socket and memory mapped file
- *phydev:* PHY implementation in Matlab
  - Primitives*:* signal processing blocks
  - L1 controller, TPU, handles: event scheduling, controlling

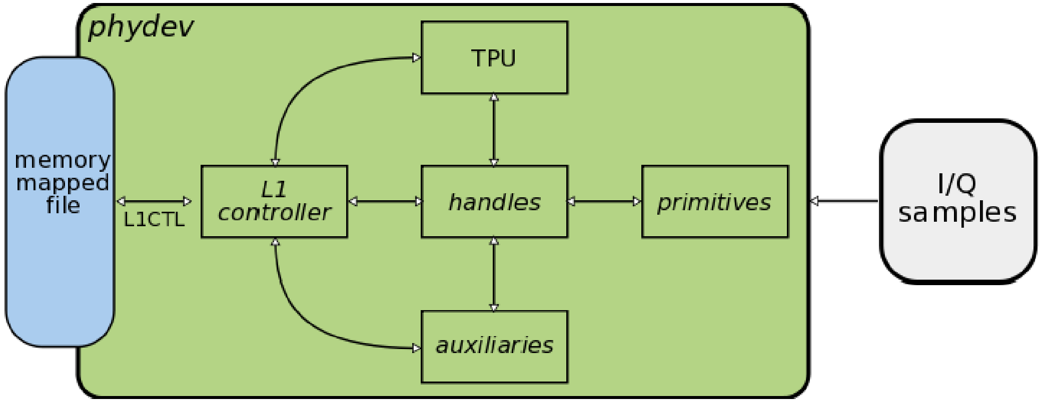


phyconnect: Interfacing OsmocomBB & Matlab

- Interfacing *mobile* (C) and *phydev* (Matlab)
- Matlab inter-process communication
  - TCP/IP socket
  - Memory mapped file
  - MEX function
- Requirements
  - Fast and simple
  - Non blocking operation
  - Best option: memory mapped file

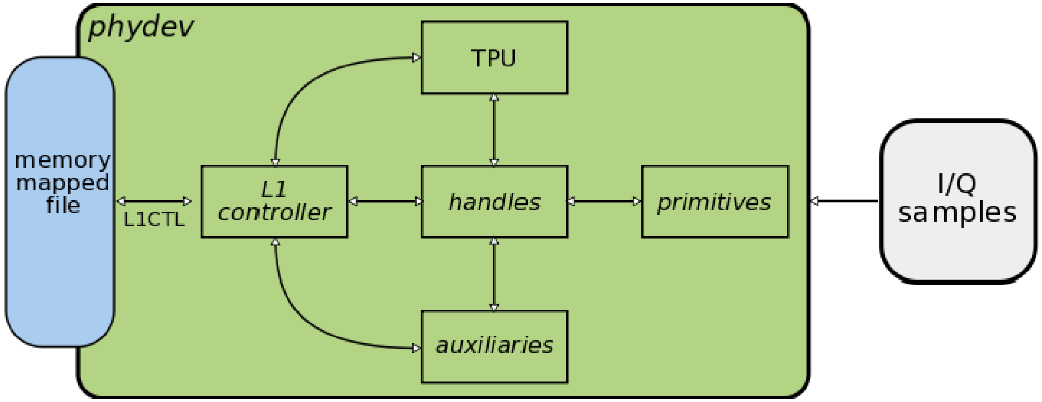
phydev: A PHY realization in Matlab for GSM

- L1 Controller
  - Dispatch L1CTL messages
- TPU
  - GSM counters, FSM according to standard
- handles
  - Controllers of receiver blocks, call and evaluate primitives



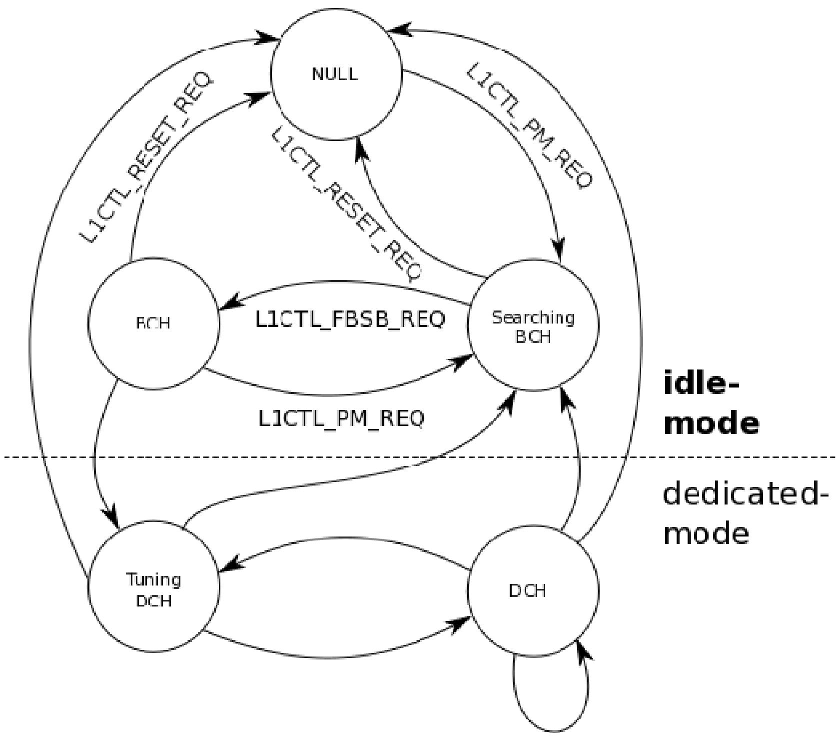
phydev: A PHY realization in Matlab for GSM

- primitives
  - Signal processing blocks
  - Operate on a defined amount of I/Q samples
- auxiliaries
  - Basic RF transceiver operations, e.g. gain settings, *tune_DCXO()*



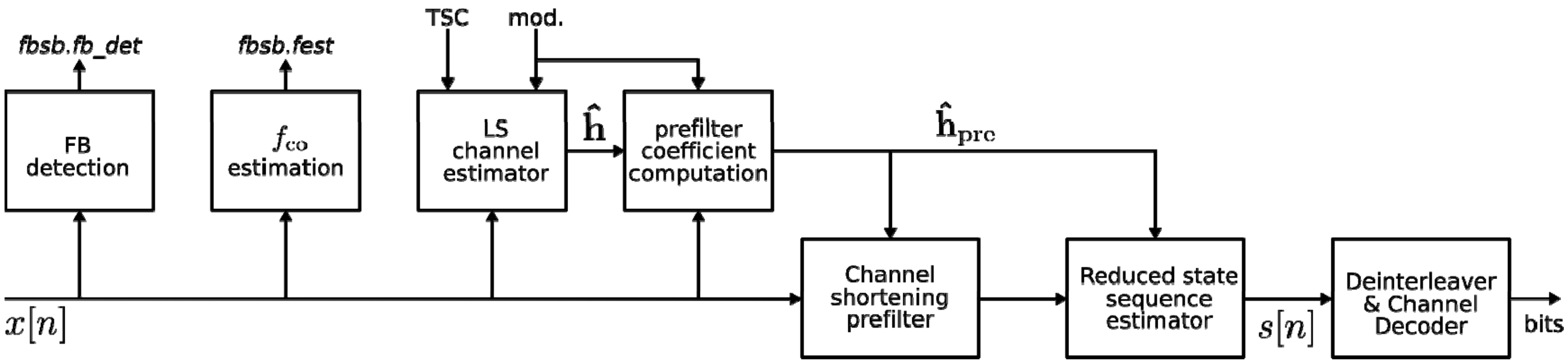
L1 Controller & TPU

- Timebase counters (QN,BN,TN,FN)
- FSM for a MS according to 3GPP TR 44.004
- Sample accurate operation
- Each primitive gets the number of samples it operates on as an argument
- Synchronization between input samples and called primitives



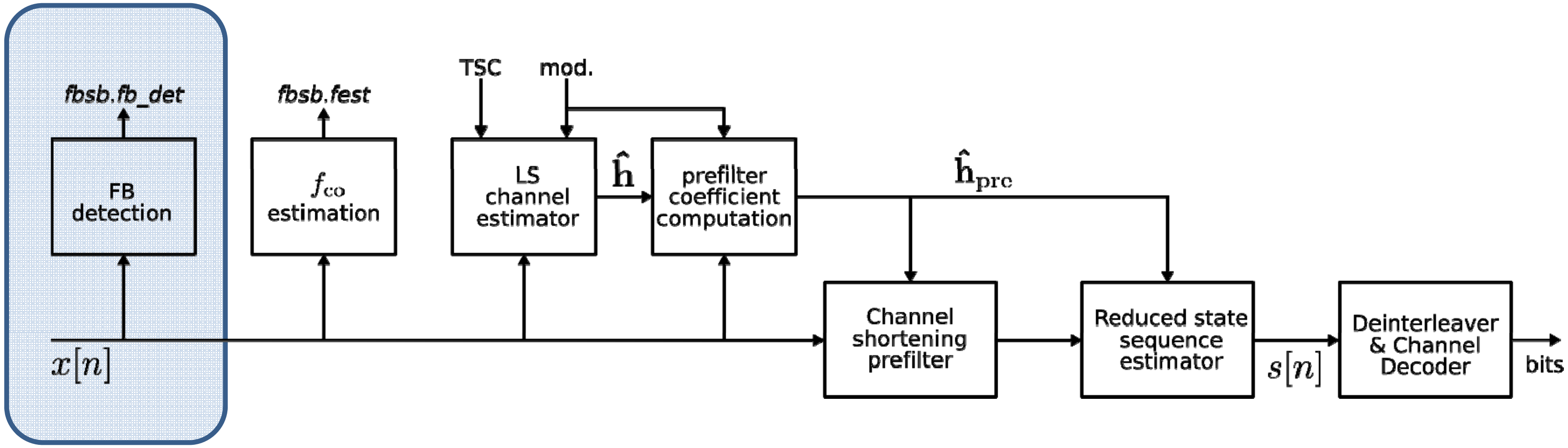
primitives: Signal Processing Blocks

- Operations on RX baseband samples e.g.
  - Frequency burst detection *FB_det()*
  - Carrier Frequency offset estimation *FB_est()*
  - Normal Burst demodulation *NB_demod()*



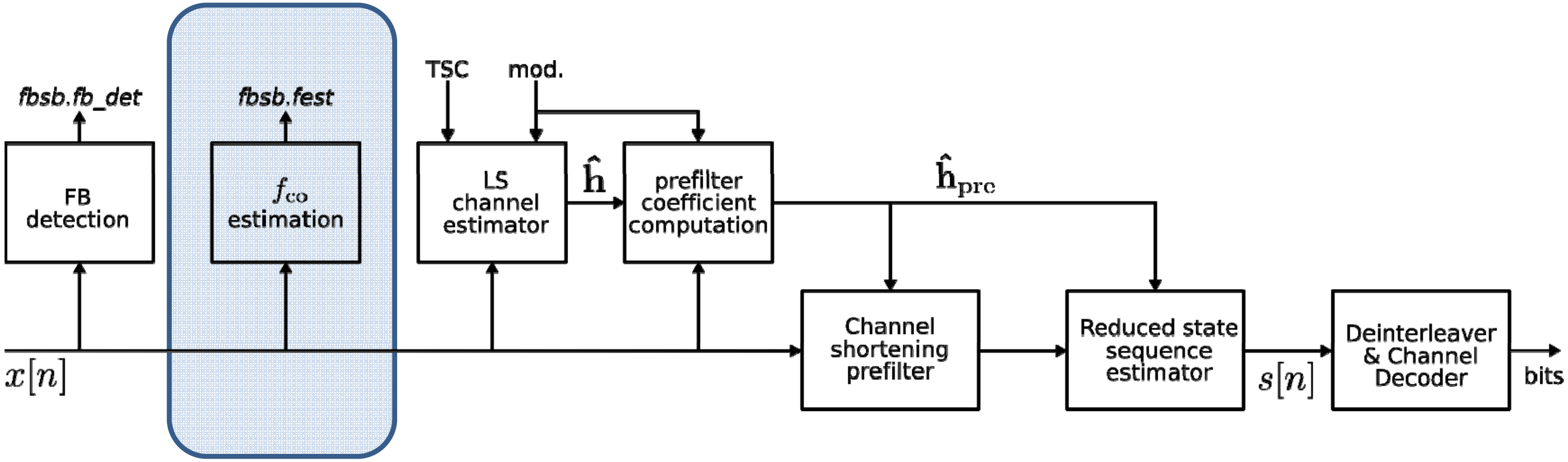
primitives: Signal Processing Blocks

- Operations on RX baseband samples e.g.
  - Frequency burst detection *FB_det()*
    - Detection of a complex sinusoid



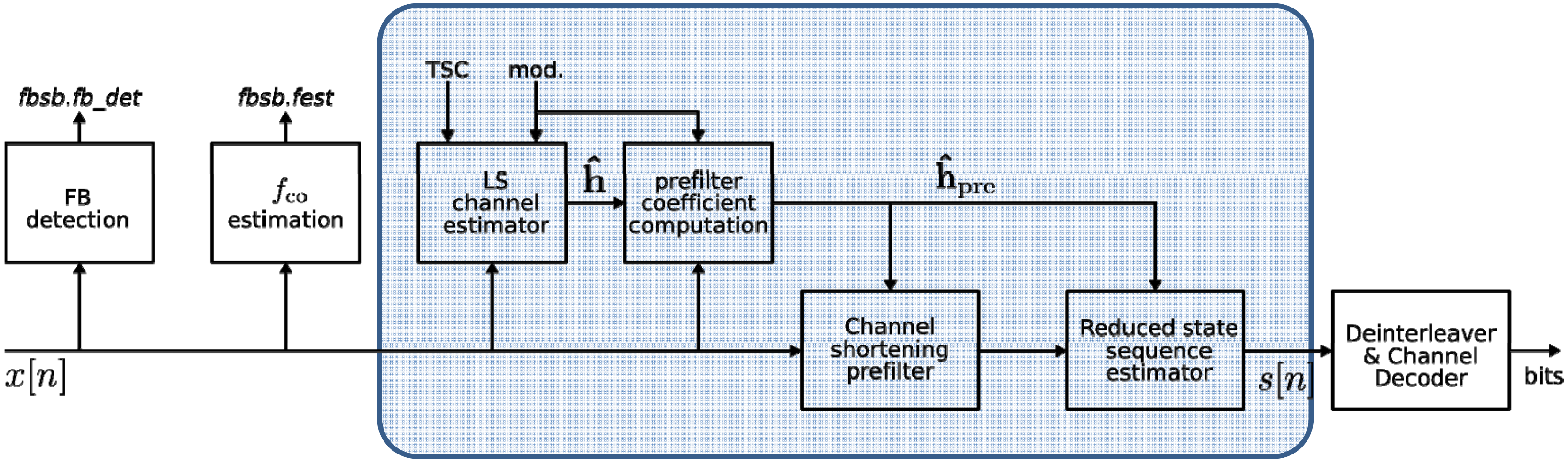
primitives: Signal Processing Blocks

- Operations on RX baseband samples e.g.
  - Carrier Frequency offset estimation *FB_est()*
    - Correlation based estimator
    - Accuracy below 0.1 ppm of carrier frequency



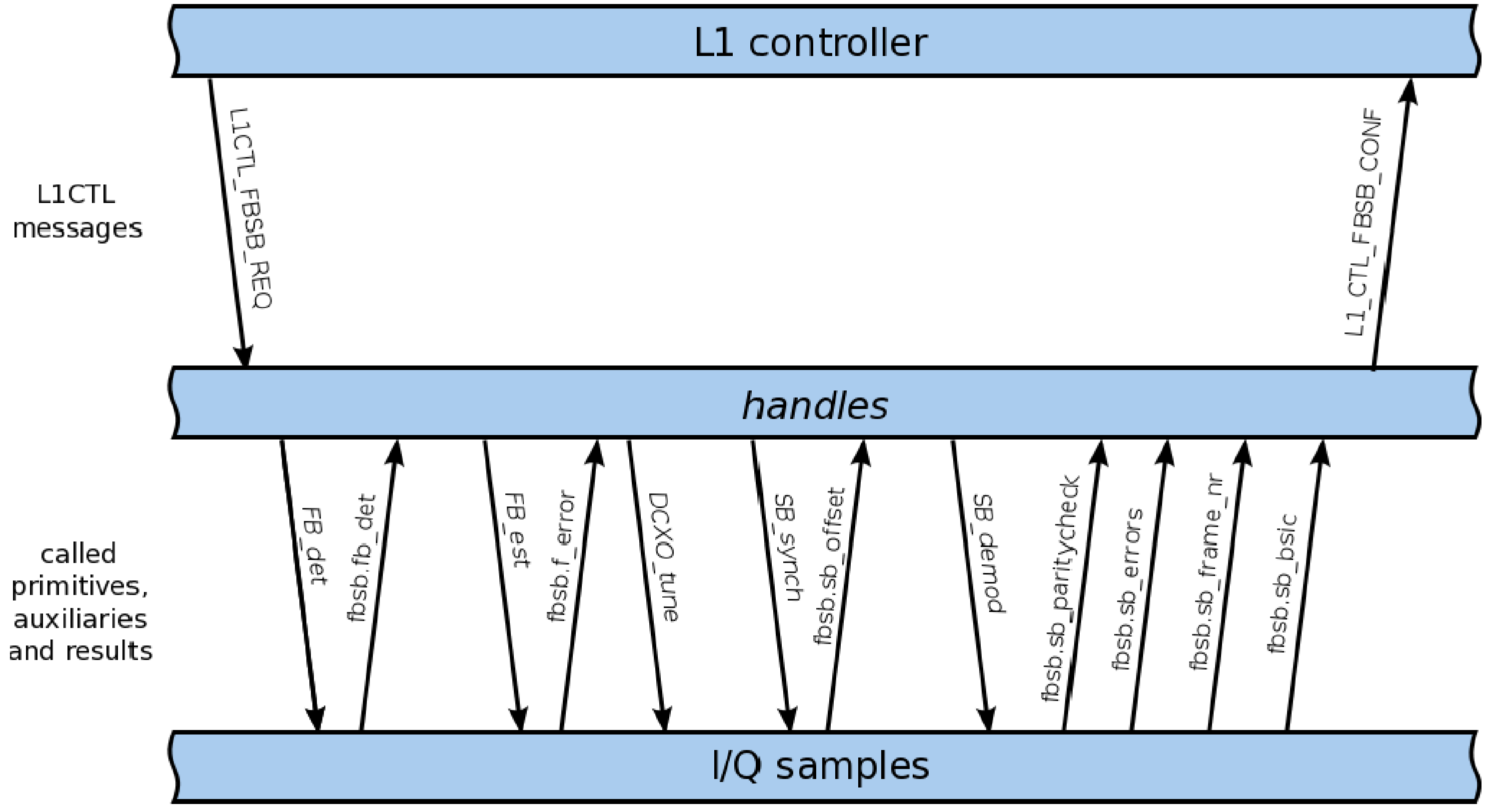
primitives: Signal Processing Blocks

- Operations on RX baseband samples e.g.
  - Normal Burst demodulation *NB_demod()*
    - Least squares channel estimator
    - Channel shortening linear filter
    - Reduced State Sequence Estimator



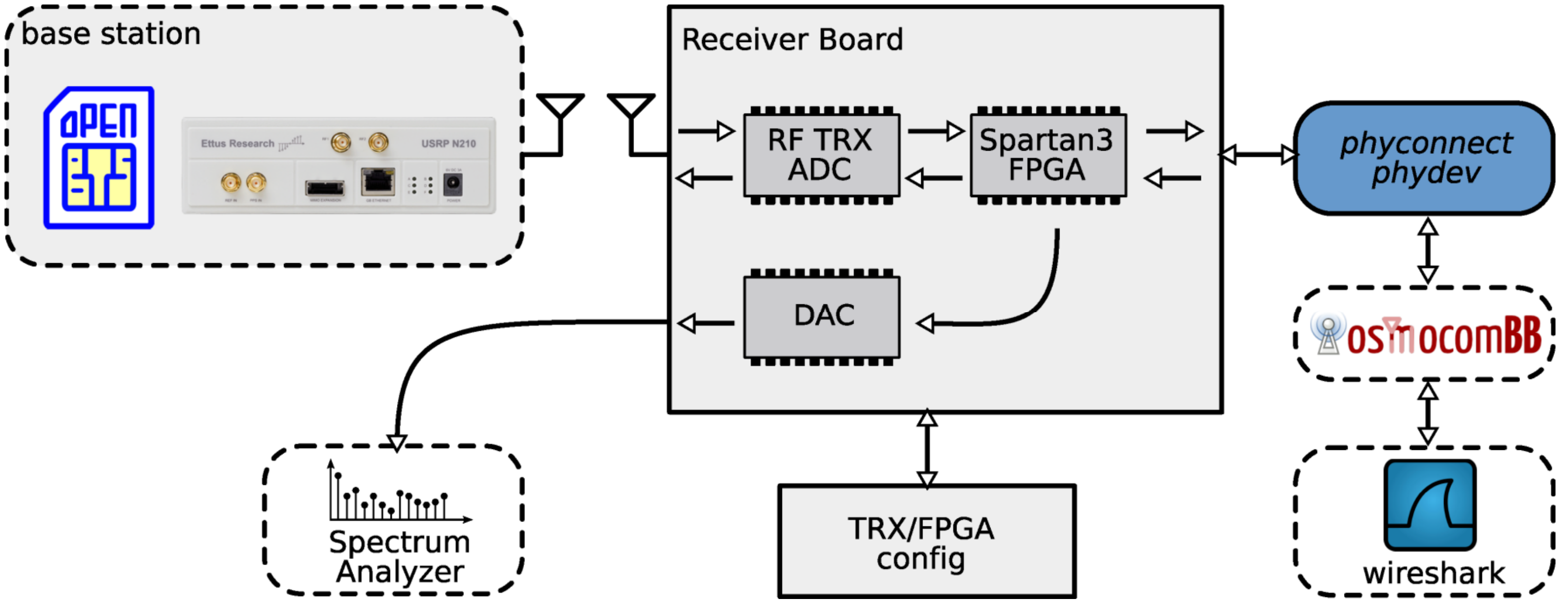
Exemplary Processing of a L1CTL Message

- Synchronization procedure: L1CTL_FBSB_REQ message



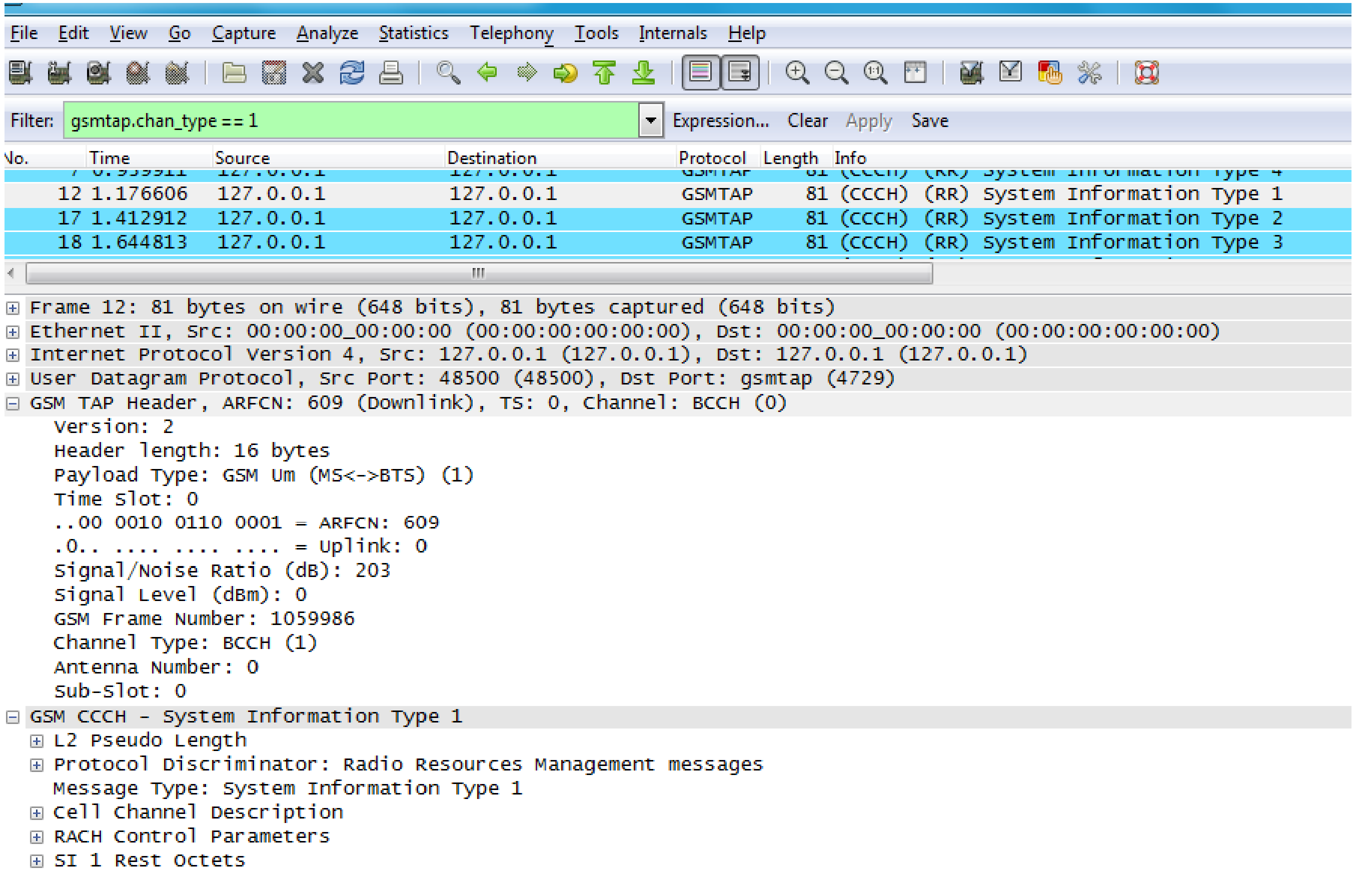
Testbed Setup

- OpenBTS as base station, wireshark for visualization
- Over the air interface
- State-of-the art multiband RF transceiver



Wireshark output, GSM system information

- GSM state: *camping on any cell*



Conclusion

- Running a complete GSM stack is fruitful for PHY algorithm development
- There is a growing interest in PHY operations, also by SDR and open source communicites
- We have shown OsmocomBB can be interfaced to PHY simulation framework

Conclusion

- Running a complete GSM stack is fruitful for PHY algorithm development
- There is a growing interest in PHY operations, also by SDR and open source communicites
- We have shown OsmocomBB can be interfaced to PHY simulation framework

Thank you for your attention!