VapsBee

Michael Lippautz Simon Kranzer Thomas Pfeiffenberger

January 28, 2010

Outline

Introduction

Goal

Real-time positioning of an embedded device using:

- Zigbee nodes placed in space
- Positioning based on signal strength indicators
- C and nesC as programming languages
- 400msec./position

Introduction (cont)

Challenges

- Programming the Beacons
- Timing constraints
- Board setup
- Hardware restrictions on signal strength (8bit value)
- Choose an adequate algorithm (positioning)

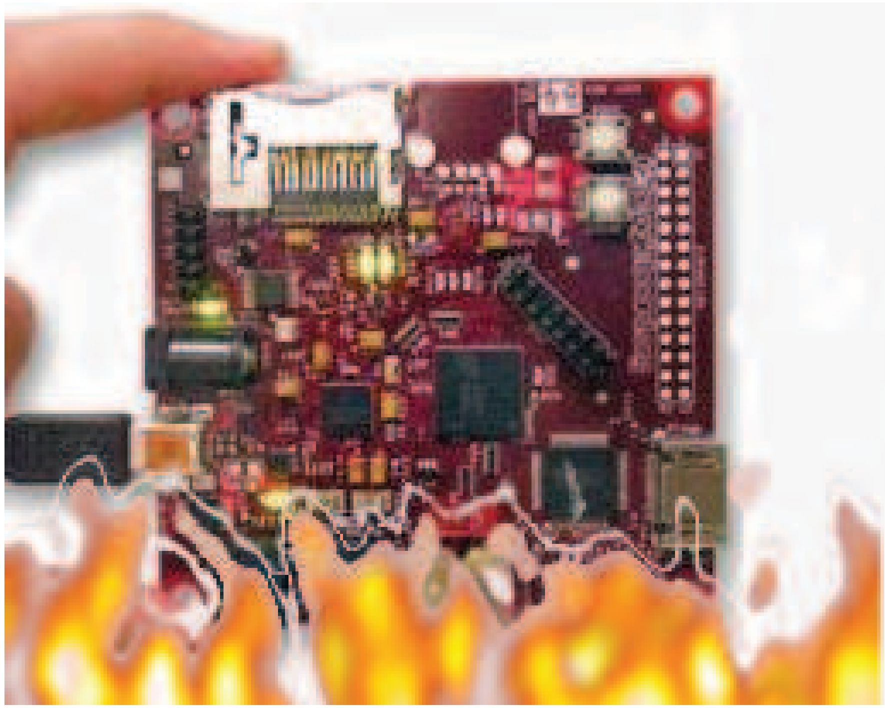
Introduction (cont)

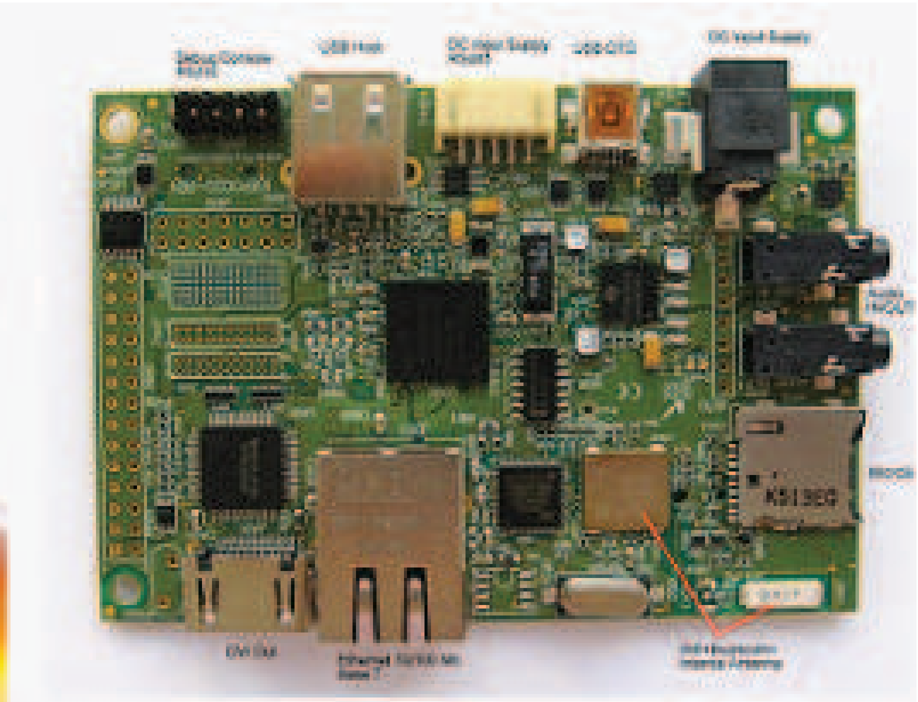
Teammembers and their jobs

- Michael Lippautz
  - Beaglebord - setup and programming
  - Igep V2 - setup and programming
  - Presentation, Dokumentation
- Simon Kranzer
  - Zigbee (sender & receiver) - setup and programming
  - Positioning with Zigbee - inquest
  - Presentation, Dokumentation
- Thomas Pfeiffenberger
  - Positioning - inquest and programming
  - Presentation, Dokumentation

Beaglboard vs. Igep V2

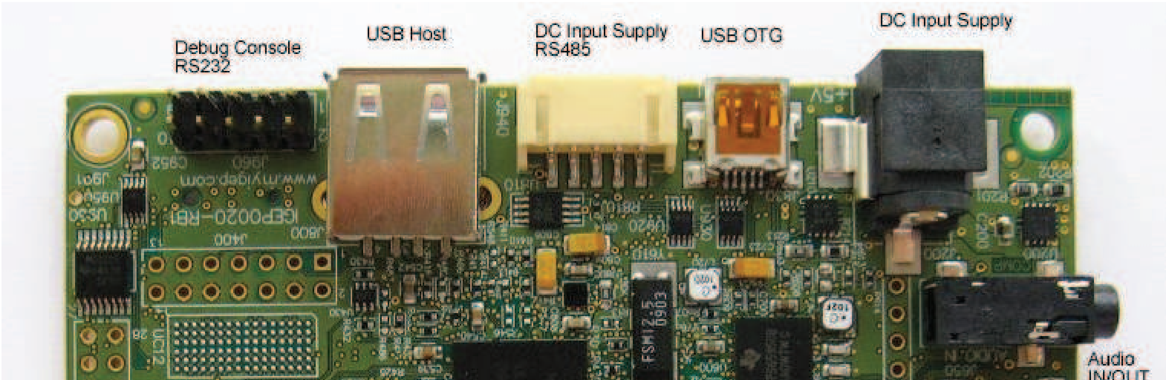
The Igep-V2-board replaces the Beagleboard since we destroyed it during the last tests by blowing the processor! ☺





Igep V2

- Single board computer (like Beagleboard)
- Designed by ISEE
- Specification (`igep-platform.com`)
- Built around OMAP3530 processor
    - ARM Cortex-A8
    - 600MHz
    - 256KB L2 cache
    - DSP TMS320C64x+
    - POWERVR SGX 530
    - Wifi IEEE 802.11b/g



MeshNetics ZDM1281-A2 (ZigBit)

Sender and Receiver

Radio

- IEEE 802.15.4 (WPAN)
- Zigbee

Chips

- ATmega1281
  - 16MHz
  - 128KB Flash
- AT86RF230 transceiver
  - 2.4GHz

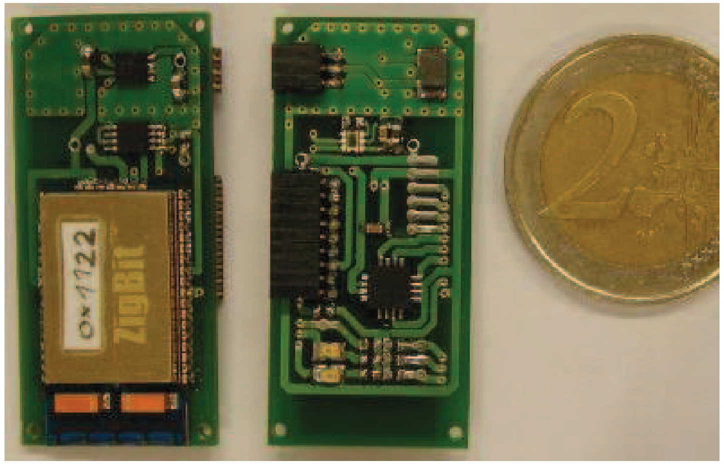
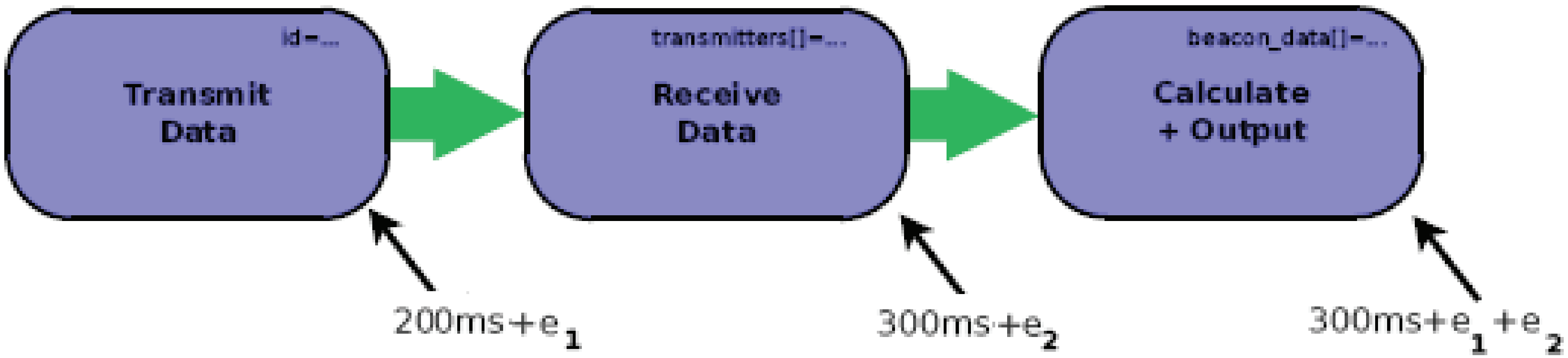


Figure: MeshNetics Zigbit

Application



Toolchain (Zigbee)

Stack

- OpenMAC
- 802.15.4 Open Source MAC Layer
- Based on TinyOS an event-driven OS designed for sensor networks with limited resources

Language

- MeshC by (`Luxoft Labs`)
- Based on NesC for TinyOS1.1

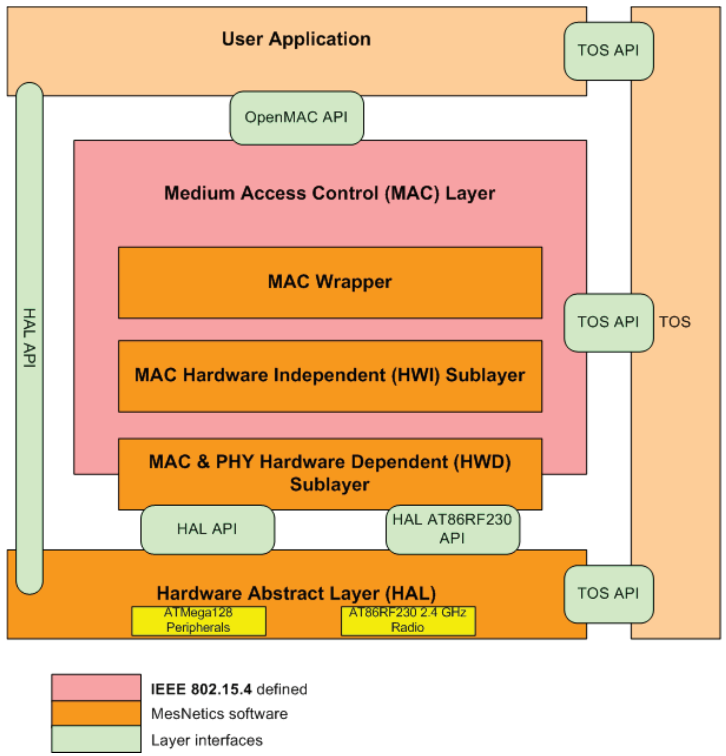


Figure: OpenMAC architecture

Sender

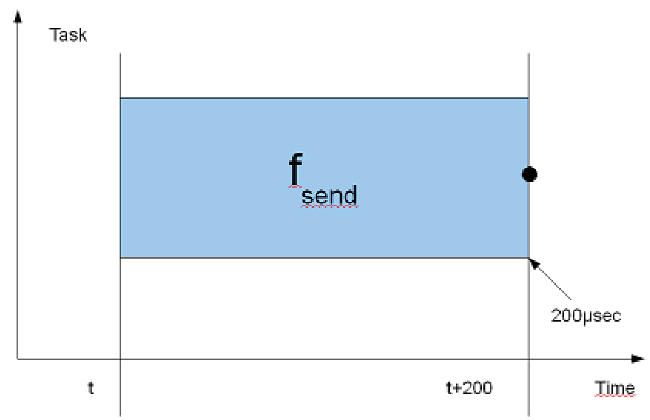
Key facts

Timing

- Send rate of $T \geq 10msec$ via ZigBee
- For the demo: $200msec$

Data

- ID
- Temperature, Light, Acceleration (not used)



Receiver

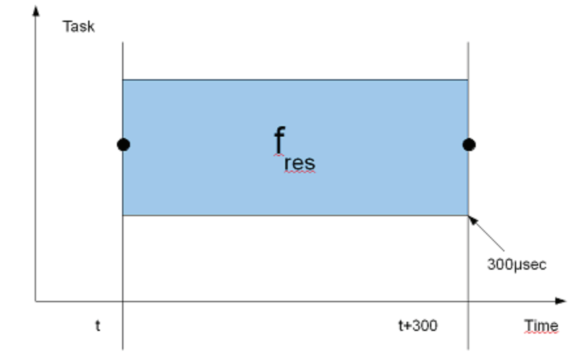
Key facts

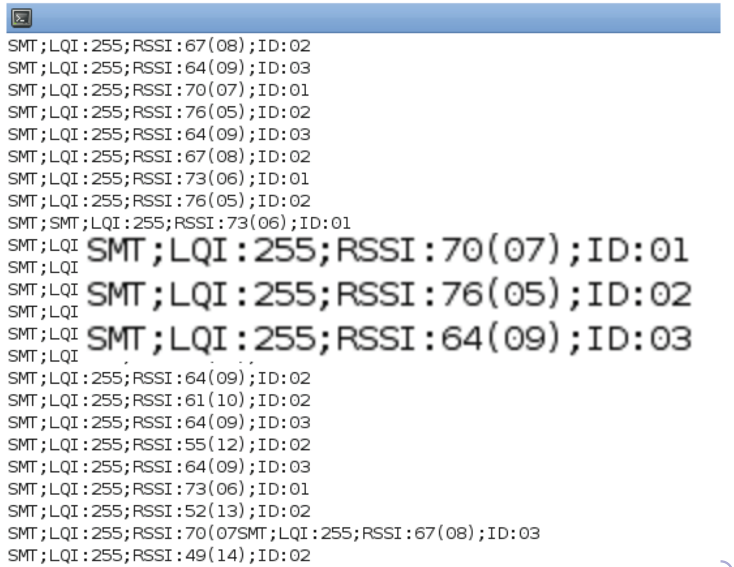
Timing

- $T \geq 50msec$ incl. processing
- For the demo: $300msec$

Dataflow (1-cycle)

- Receive packets from nodes
- Get IDs from packets
- Aquire RSSI and LQI (Stack)
- Calculate packetRSSI, averagedRSSI (variable)
- Send data via UART





General (Board)

- Kernel and rootfs built using OpenEmbedded (`openembedded.net`)
- Distribution: Ångström (with reduced # of packages)

Problems

- Standard Linux Kernel preemption (no RT at all)
  - When CPU is in user-mode
  - When Kernel returns to user space from syscall
  - When Kernel is locked by mutex
- Only 1 explicit serial port available

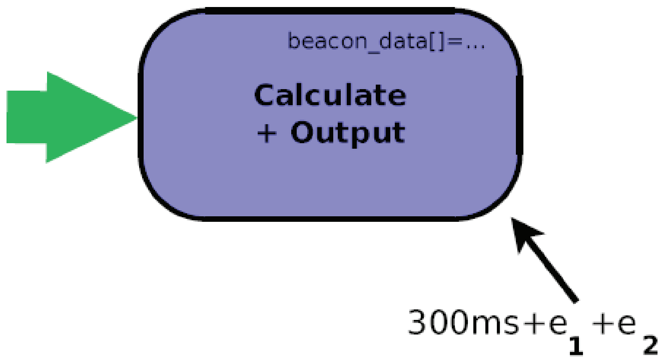
Solutions

RT

Apply official RT patch
(`rt.wiki.kernel.org`)

- **NO** hard real-time guarantees ☹
- *BUT*
  - Kernel preemptable (mutex/spinlocks and interrupts)
  - Priorities on programs

$\Longrightarrow$ 1 task RT manageable



Serial port

/* Use Kernel based pin muxing to enable an additional UART (beagle) */

Main program

1. Initialization

```
set scheduling priority // 1 above kernel
/* open & configure serial port (beagle) // data source */
open & configure usb2serial port (igep)
acquire memory // use to store node data
lock heap // no memory alloc after this point
```

2. Be RT

```
while (TRUE)
  {
        sleep [xxx us - xx ms]
        <<threaded>>
        read data into memory
        calc position
        output
  }
```

Positioning

Challenges

- Triangulation
  - measuring angles
- Trilateration
  - measuring distances
- Alternativ Positioning Systems
  - Point of interest
  - Human interaction
- Choose an adequate algorithm $\Rightarrow$ Trilateration

Trilateration

Distance calculation

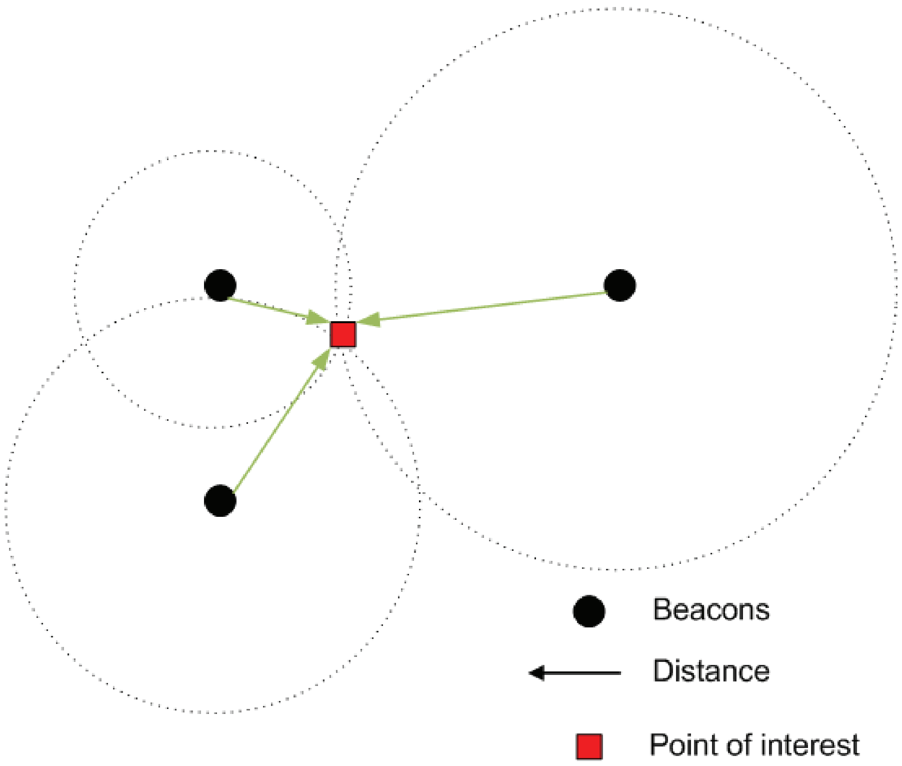


Figure: Trilateration

Trilateration

Distance calculation

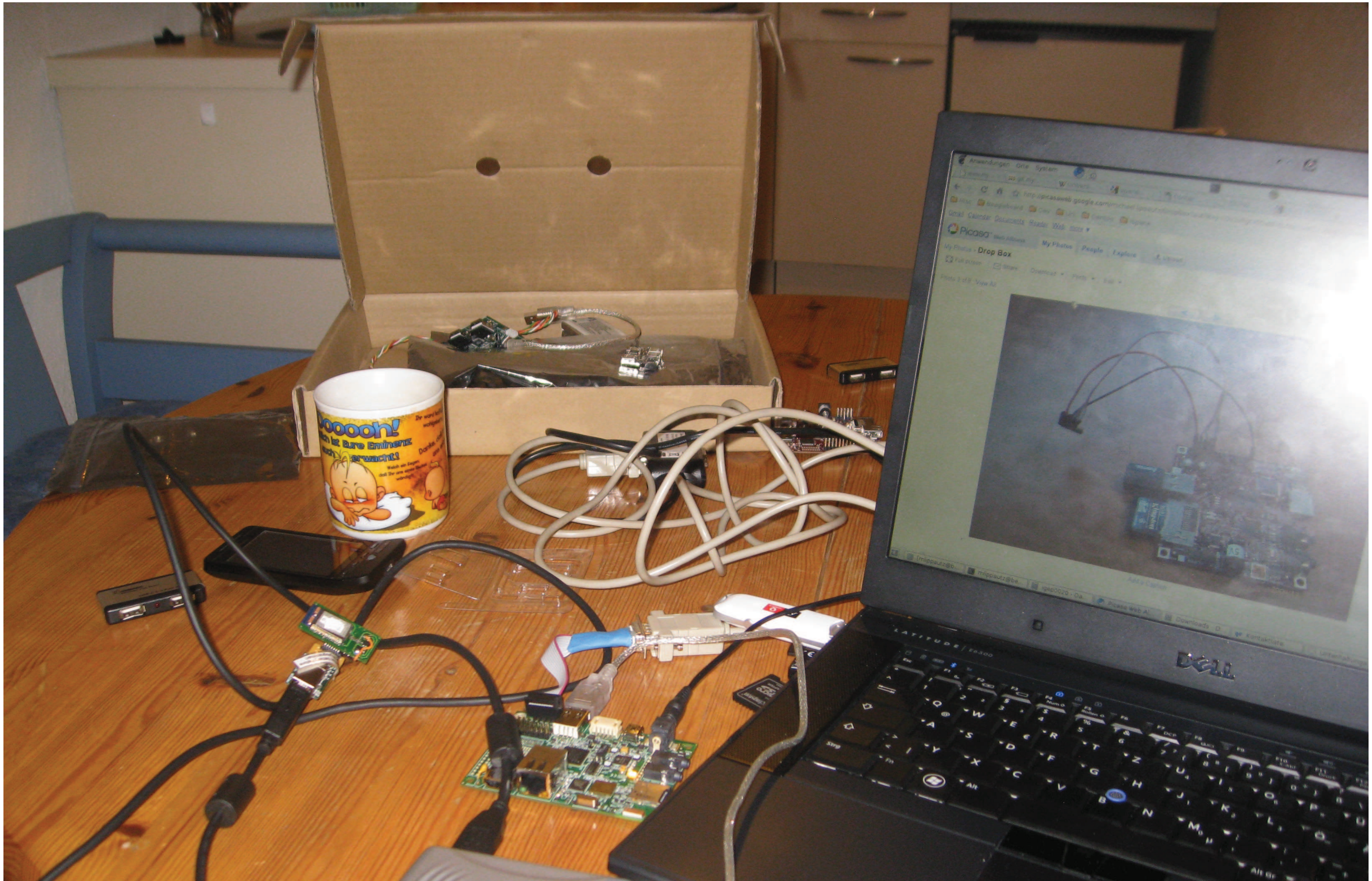
- RSSI Received Signal Strength Indication
- Mapping RSSI $\Longleftrightarrow$ Distance
- LQI Link Quality Indication
- $RSSI(d) = RSSI(d_0) + 10n * \log(\frac{d}{d_0})$
- $dist_A^2 = (x_0 - x_A)^2 + (y_0 - y_A)^2$

Conclusion

Open issues

- Hardware restrictions on signal strength (8bit value)
- Evaluation of the positioning algorithm
- no hard real time -> GIOTTO e-machine
- defining new Use Case "'**Guaranteed measurements**"' (e.g. temperature)

THANK YOU!



Positioning with Zigbee

Measurements

- RSSI - Received Signal Strength Indicator
- LQI - Link Quality

Calculation

- Beacon broadcast its position via Zigbee
- Distance by RSSI or LQI
- Position by Weighted Centroid Localization (WCL)[1]

$P_i(x, y) = \frac{\sum_n^{j=1}(w_{ij}B_j(x,y))}{\sum_n^{j=1} w_{ij}}$ where P_i is a position, B_j are the beacons and w_{ij} with $w_{ij} = \frac{1}{(d_{ij})^g}$ are the weights and d_{ij} is the (measured) distance to a beacon and g the degree to ensure remote beacons have still impact.

[1] Localization in Zigbee-based Sensor Networks, Jan Blumenthal et al., WISP 2007