Die Wiselib und iOS

Marcus Brandenburger

// Today

Inhalt

- iOS Geräte
- Idee
- iOS Prgrammierung
- Wiselib External Interfaces
- Live Demo

iOS Geräte

- iPod touch
- iPad
- iPhone

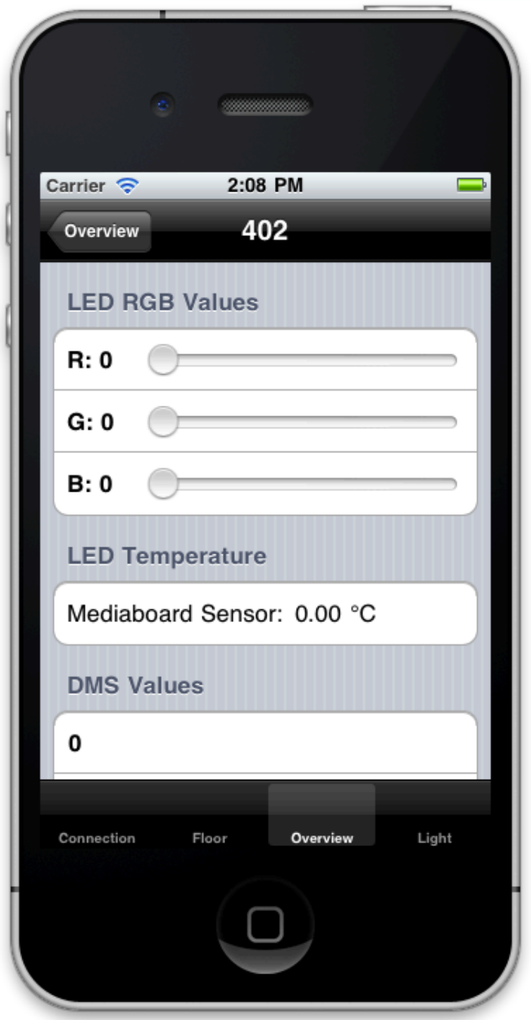
iOS Geräte

- Sensoren
  - 3-Achsen-Gyrosensor
  - Beschleunigungssensor
  - Umgebungslichtsensor
- Multi-Touch-Widescreen Display
  - 960 x 640 bzw. 1024x 768
- Drahtlose Technologie
  - Wi-Fi (802.11a/b/g/n)
  - Bluetooth 2.1 + EDR
- Sonst so
  - 1-2 Kameras, 8-64 GB Speicher

mehr auf apple.de

iOS Simulator

- iOS Simulator (iPhone und iPad)
- keine Beschleunigungs-/Kamera Simulation
- Rotieren, Schütteln, Touch-Events
- GPS Koordinaten aus dem Apple Headquarter
- Trotzdem cool zum Entwickeln



Idee


Carrier
2:08 PM
Xcode4Test
Minesweeper
wisePhone
wisePad
MiniFl...ontrol
Wiseli...rTest
Contacts
Safari

Problem

802.11n

802.15.4

Lösung:
Wiselib und die Testbed-Runtime

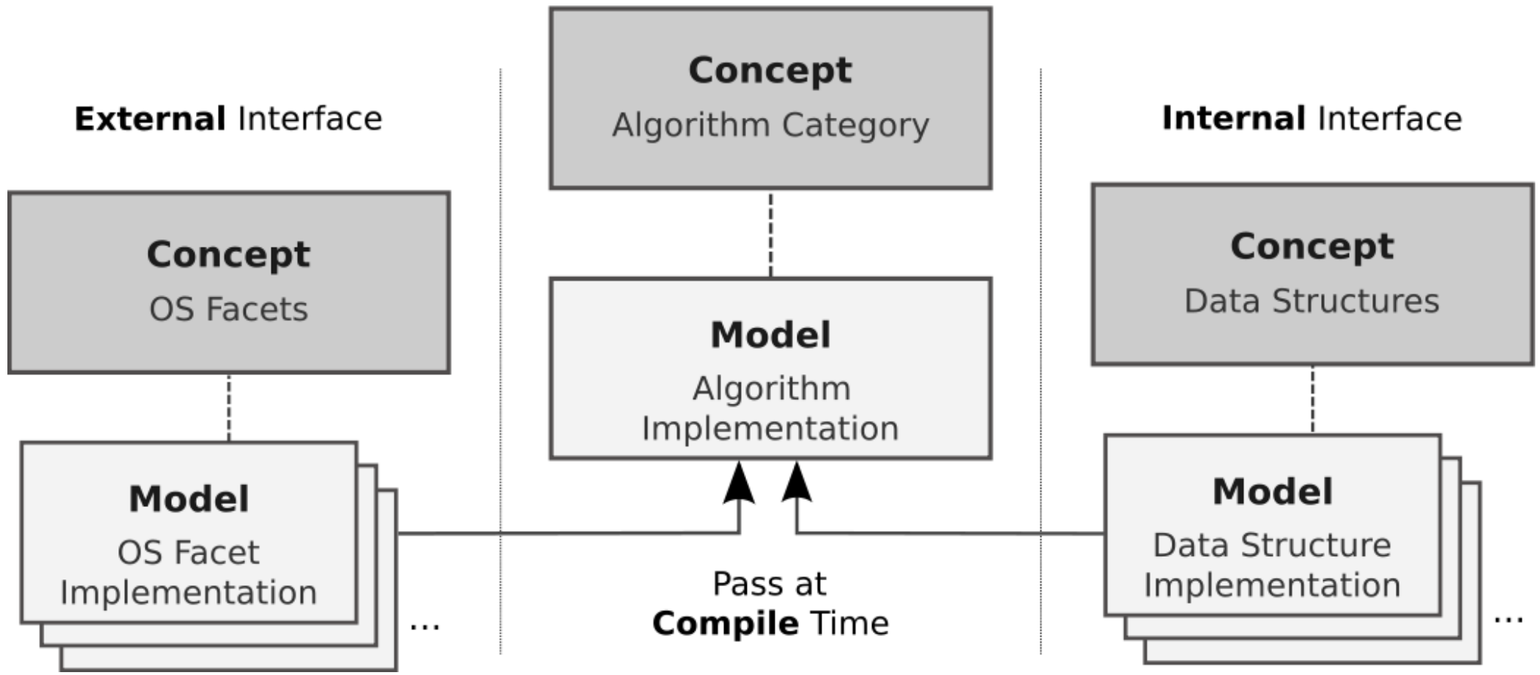
Testbed-Runtime

iOS Programmierung

- iOS SDK + XCode IDE über iTunes
- Objectiv-C 2.0 / C++ / C
- Debugging
  - Simulator
  - iPhone, iPod, iPad (Entwicklerlizenz nötig)

Wiselib für iOS

- iOS External Interface
  - Os Facet
  - Radio
  - Timer
  - Debug



iOS Radio Model

- `int enable_radio()`
- `int disable_radio()`
- `int send(node_id_t receiver, size_t len, block_data_t *data )`
- `node_id_t id()`
- `int reg_recv_callback(T *obj_pnt )`
- `int unreg_recv_callback(int idx )`
- `void set_testbed(NSString *url, UInt16 port)`
- `private: iOSRadioController *iOSRadio_`

iOSRadioController

- Socketverbindung zur Testbed-Runtime
- Sendet / Empfängt Daten (iOS CoreServices)
- Verpackt/Entpackt Nachrichten in das TR Protocol Buffers Format

Wiselib Application Demo

- Live Demo

Danke!