

ALG-Seminar Sensornetze

Verteilte Plattenerkennung

Tobias Baumgartner

TU Braunschweig

18. Mai 2011

Outline

- 1 Setup
 - Großer Flur
 - Miniflur
- 2 Sensorerkennung
- 3 Plattenerkennung
 - Platte belastet
 - Platten-IDs

Outline

1 Setup

Großer Flur

Miniflur

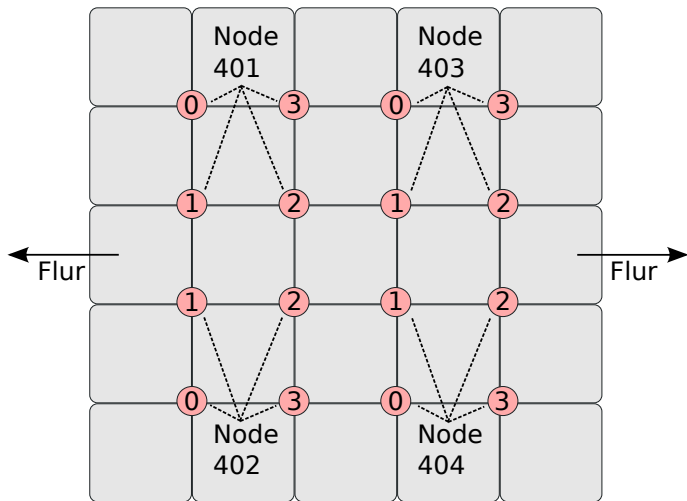
2 Sensorerkennung

3 Plattenerkennung

Platte belastet

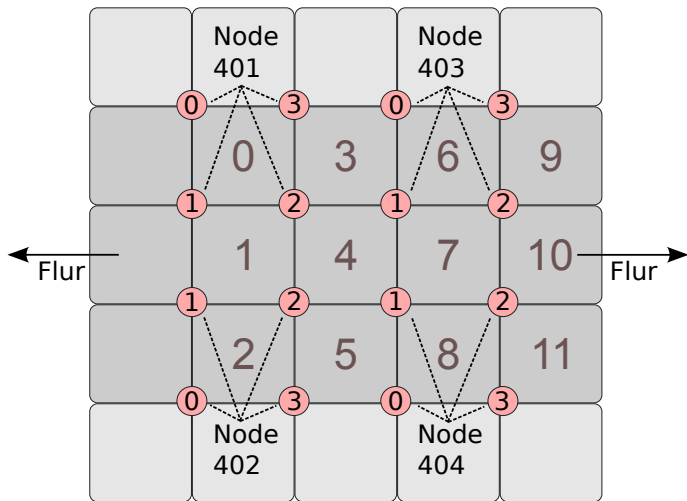
Platten-IDs

Sensorplatzierung



● = DMS Sensor

Sensorplatzierung



● = DMS Sensor

Outline

1 Setup

Großer Flur

Miniflur

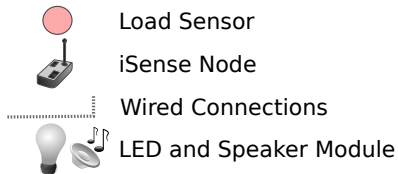
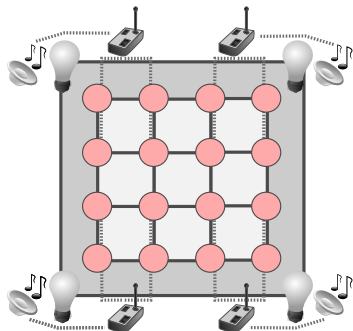
2 Sensorerkennung

3 Plattenerkennung

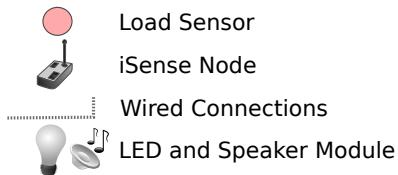
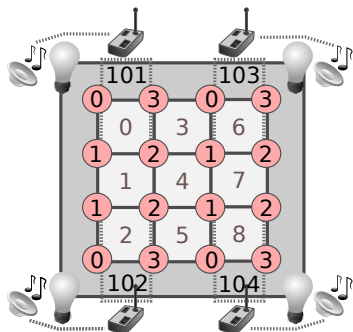
Platte belastet

Platten-IDs

Sensorplatzierung



Sensorplatzierung



Outline

- 1 Setup
 - Großer Flur
 - Miniflur
- 2 Sensorerkennung
- 3 Plattenerkennung
 - Platte belastet
 - Platten-IDs

Basics

- Funktionsweise wie bei Algorithmen Praktikum
 - *Moving Average*
 - Standardabweichung
 - *Moving Average* wird nur gebildet, wenn kein (potentielles) Event detektiert ist
- Rohwerte: 250 Hz
- Array von 300 Werten/DMS ($300 * 4 * 2 = 2400$ Byte auf iSense!)
- 4 potentielle Events für Sensorevent

Sensorevent (1 von 2)

- Aktualisiere Sensor-ID in Event-Liste
 - Timestamp (*lokale Zeit*)
 - Setze Event auf `true` (*Paar: lokale Knoten-ID, Sensor-ID*)
- Wenn in den letzten 200ms ein Event ausgelöst wurde
⇒ Abbruch
sonst
⇒ Starte Timer `SEND_WAIT` (50ms)
- Starte Timer: `EVALUATION_WAIT` (100ms) → Plattenerkennung
- Nach Timerablauf `SEND_WAIT`
 - Fasse alle lokalen Sensor-Events der letzten 55ms in Nachricht zusammen
 - Broadcast der Nachricht

Sensorevent (2 von 2)

- Empfang von Nachricht mit Sensor-Events
 - Aktualisiere Sensor-ID in Event-Liste
 - Timestamp (*Empfang der Nachricht*)
 - Setze Event auf true (*Paar: Knoten-ID Sender, Sensor-ID*)
 - Starte Timer: EVALUATION_WAIT (100ms) → Plattenerkennung
- Aufräumen von Sensor-Events
 - Timer CLEANUP_WAIT alle 500ms im Hintergrund
 - Lösche alle Events, die älter als EVENT_EXPIRATION (300ms) sind

Outline

1 Setup

- Großer Flur
- Miniflur

2 Sensorerkennung

3 Plattenerkennung

- Platte belastet
- Platten-IDs

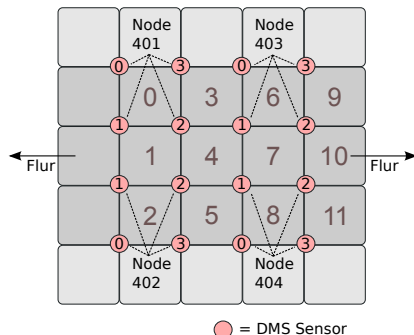
Basics

- Sensor-Events werden per Funk an Nachbarn verteilt
- Löschen $\geq 300\text{ms}$, neu verschicken alle 200ms
- Ein Platte ist belastet, wenn mind. 3 Sensoren der Platte belastet sind
- **Jeder** Knoten überprüft **alle** angrenzenden Platten
- Es gibt **keine** *Platte-nicht-mehr-belastet*-Events

Outline

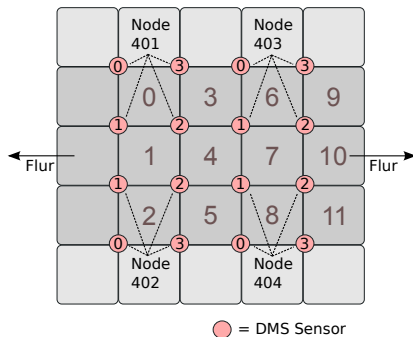
- 1 Setup
 - Großer Flur
 - Miniflur
- 2 Sensorerkennung
- 3 Plattenerkennung**
 - Platte belastet
 - Platten-IDs

Platte-belastet-Event (1 von 3)



- Lokal
 - Alle Sensoren des lokalen Knotens

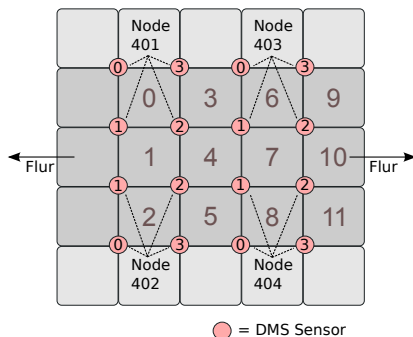
Platte-belastet-Event (1 von 3)



- Links

- Sensoren 0, 1 des lokalen Knotens
- Sensoren 2, 3 des Nachbarn mit lokaler ID - 2

Platte-belastet-Event (1 von 3)



- Gegenüber

- Sensoren 1, 2 des lokalen Knotens
- Sensoren 1, 2 des gegenüberliegenden Knotens A
- $A = \text{myID} \% 2 == 0 ? \text{myID} - 1 : \text{myID} + 1$

Outline

1 Setup

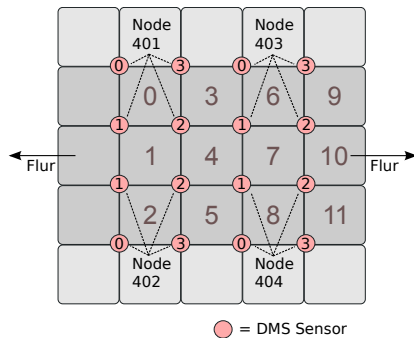
Großer Flur
Miniflur

2 Sensorerkennung

3 Plattenerkennung

Platte belastet
Platten-IDs

Lokale Platten-ID



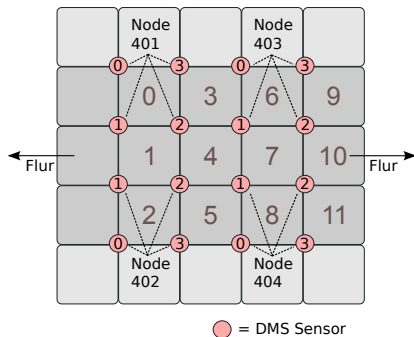
- Berechnung von lokaler Platten-ID einfach möglich :)

```

1 | // get node ids 1..x for both 101..104 and 401..430
2 | node_id = node_id % 100;
3 | int row = ((node_id - 1) / 2) * 6;
4 | // offset is 0 for nodes with odd ids, 2 for nodes with even i
5 | int offset = ((node_id - 1) % 2) * 2;
6 | return row + offset;

```

Globale Platten-IDs



- *myID* + gegenüberliegende ID / 2
- Beispiele
 - Links neben 404: $(8 + 2) / 2 = 5$
 - Gegenüber von 403: $(8 + 6) / 2 = 7$
 - Rechts neben 401: $(0 + 6) / 2 = 3$
 - ...

#fin

- thx!