

PG 595: Ressourcenmodellierung für Peripherie in eingebetteten Systemen

Daniel Friesel

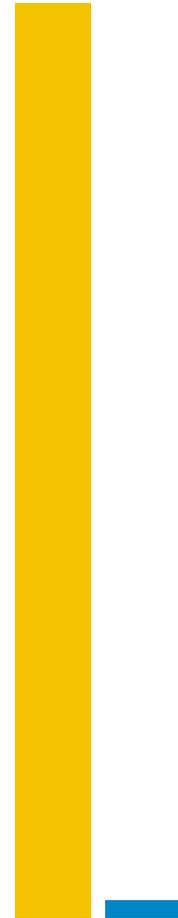
Arbeitsgruppe ESS
Informatik 12
TU Dortmund

17. Oktober 2015

- Solar Doorplate-Akku: 15 mAh bei 3,7V.

- Solar Doorplate-Akku: 15 mAh bei 3,7V.
 - D.h.: ca. 30 Tage CPU+Display idle

- Solar Doorplate-Akku: 15 mAh bei 3,7V.
 - D.h.: ca. 30 Tage CPU+Display idle
 - Oder: ≥ 12 Stunden CPU aktiv (taktabhängig)



- Solar Doorplate-Akku: 15 mAh bei 3,7V.
 - D.h.: ca. 30 Tage CPU+Display idle
 - Oder: ≥ 12 Stunden CPU aktiv (taktabhängig)
 - Oder: ca. 30 Minuten Daten senden/empfangen



- Solar Doorplate-Akku: 15 mAh bei 3,7V.
 - D.h.: ca. 30 Tage CPU+Display idle
 - Oder: ≥ 12 Stunden CPU aktiv (taktabhängig)
 - Oder: ca. 30 Minuten Daten senden/empfangen
- Abwägung Qualität \leftrightarrow Lebensdauer



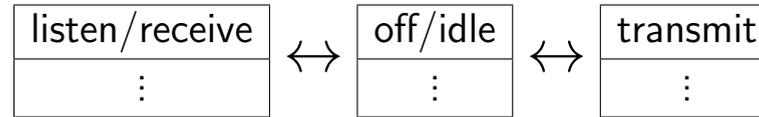
- Funkmodul, Display, ggf. Kartenleser

- Funkmodul, Display, ggf. Kartenleser
- Wenig Zustände, Verhalten bekannt → Modellierbar

- Funkmodul, Display, ggf. Kartenleser
- Wenig Zustände, Verhalten bekannt → Modellierbar
- Entwurf: Machbarkeit/Energiereserven bei typischer Nutzung:
Offline Modeling

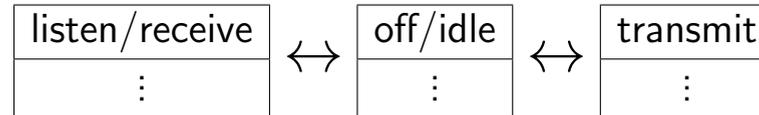
- Funkmodul, Display, ggf. Kartenleser
- Wenig Zustände, Verhalten bekannt → Modellierbar
- Entwurf: Machbarkeit/Energiereserven bei typischer Nutzung:
Offline Modeling
- Betrieb: Energie(verbrauch) kennen, ggf. Notfallplan anwenden:
Online Modeling

Offline Modeling



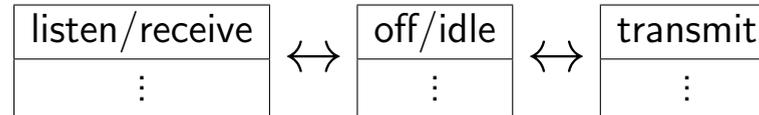
- Externes Modell, losgelöst vom Betriebssystem

Offline Modeling



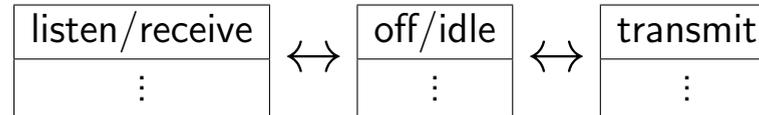
- Externes Modell, losgelöst vom Betriebssystem
- Beliebig komplex

Offline Modeling



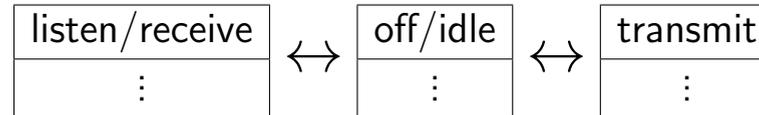
- Externes Modell, losgelöst vom Betriebssystem
- Beliebig komplex
- Simulation → Evaluation von Chips, Funkprotokollen etc.
 - Auf Basis manuell erstellter Szenarien [TRJ02] [DMC03]

Offline Modeling



- Externes Modell, losgelöst vom Betriebssystem
- Beliebig komplex
- Simulation → Evaluation von Chips, Funkprotokollen etc.
 - Auf Basis manuell erstellter Szenarien [TRJ02] [DMC03]
 - Prinzipiell sogar Model Checking [Kwi07] [VS08]

Offline Modeling



- Externes Modell, losgelöst vom Betriebssystem
- Beliebig komplex
- Simulation → Evaluation von Chips, Funkprotokollen etc.
 - Auf Basis manuell erstellter Szenarien [TRJ02] [DMC03]
 - Prinzipiell sogar Model Checking [Kwi07] [VS08]
- Betriebssystem hat keine Energiekenntnis
 - Szenarien müssen Realität gut abdecken

```
enum radio_state { OFF, IDLE, RECEIVE, TRANSMIT };  
int energy_send(radio_state prev_state, int tx_power, int length) { ... }
```

- Modell im Betriebssystem (Treiber)

```
enum radio_state { OFF, IDLE, RECEIVE, TRANSMIT };
```

```
int energy_send(radio_state prev_state, int tx_power, int length) { ... }
```

- Modell im Betriebssystem (Treiber)
- Energiezustand jederzeit bekannt
 - Unabhängig von (ggf. nicht vorhandenen) Sensoren

```
enum radio_state { OFF, IDLE, RECEIVE, TRANSMIT };
```

```
int energy_send(radio_state prev_state, int tx_power, int length) { ... }
```

- Modell im Betriebssystem (Treiber)
- Energiezustand jederzeit bekannt
 - Unabhängig von (ggf. nicht vorhandenen) Sensoren
- Adaptives Verhalten

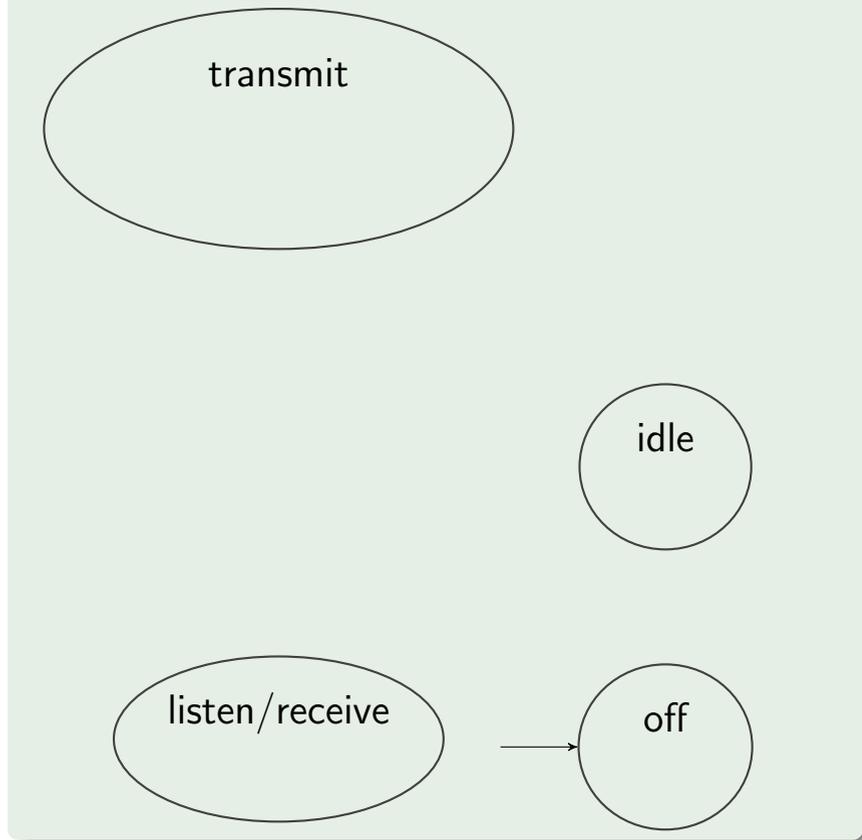
```
enum radio_state { OFF, IDLE, RECEIVE, TRANSMIT };
```

```
int energy_send(radio_state prev_state, int tx_power, int length) { ... }
```

- Modell im Betriebssystem (Treiber)
- Energiezustand jederzeit bekannt
 - Unabhängig von (ggf. nicht vorhandenen) Sensoren
- Adaptives Verhalten
- Berechnung im Treiber → Nicht zu komplex

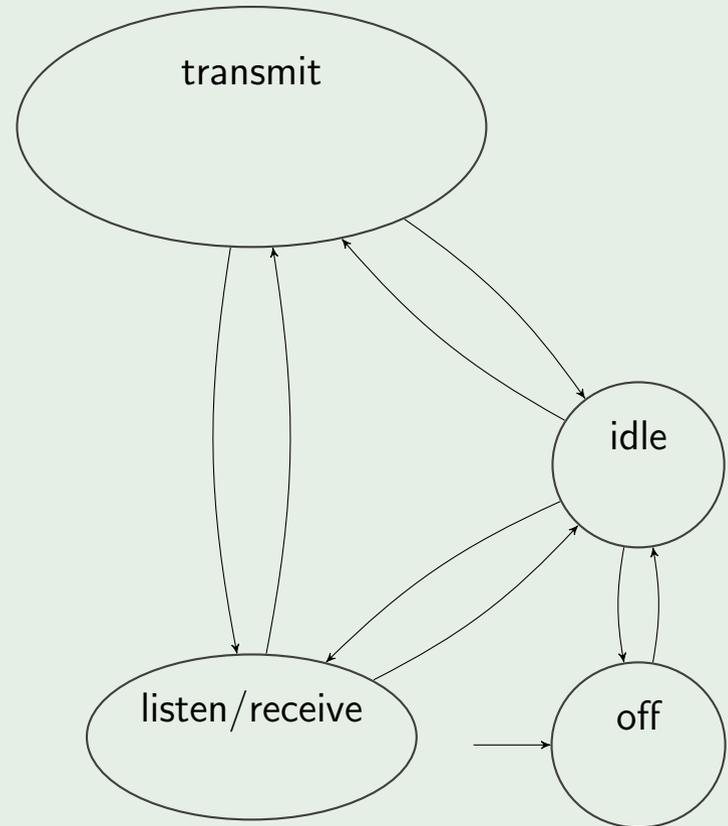
Modellierung mit Zeitautomaten

Beispiel (CC1200 @ 50kbps)



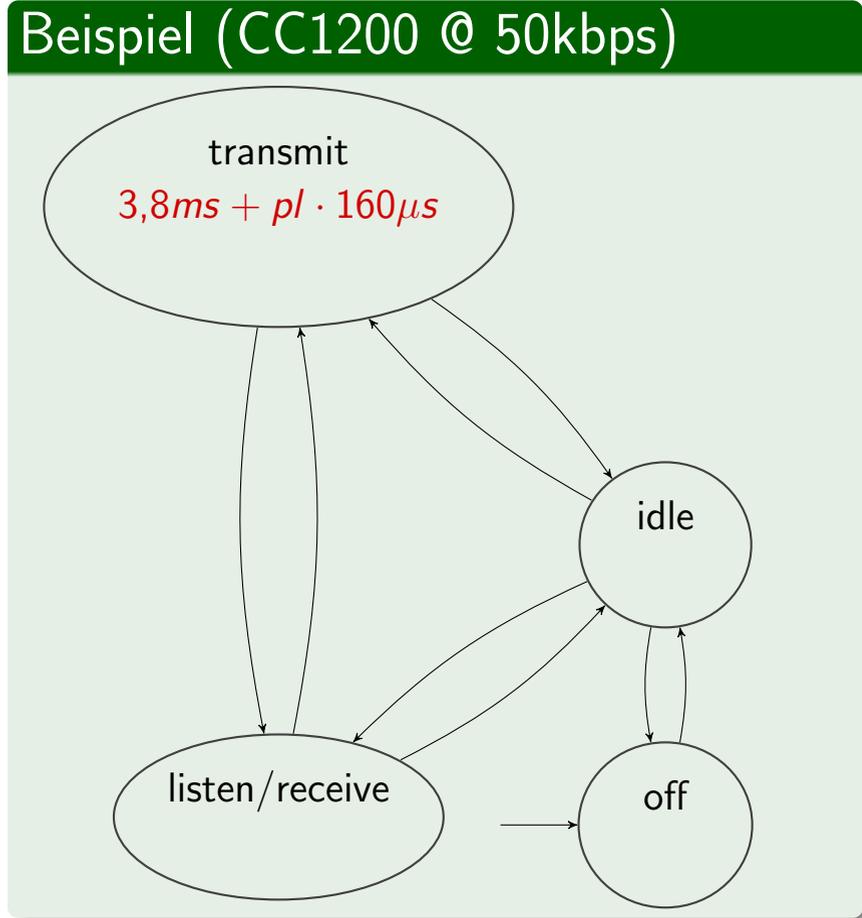
Modellierung mit Zeitautomaten

Beispiel (CC1200 @ 50kbps)



Modellierung mit Zeitautomaten

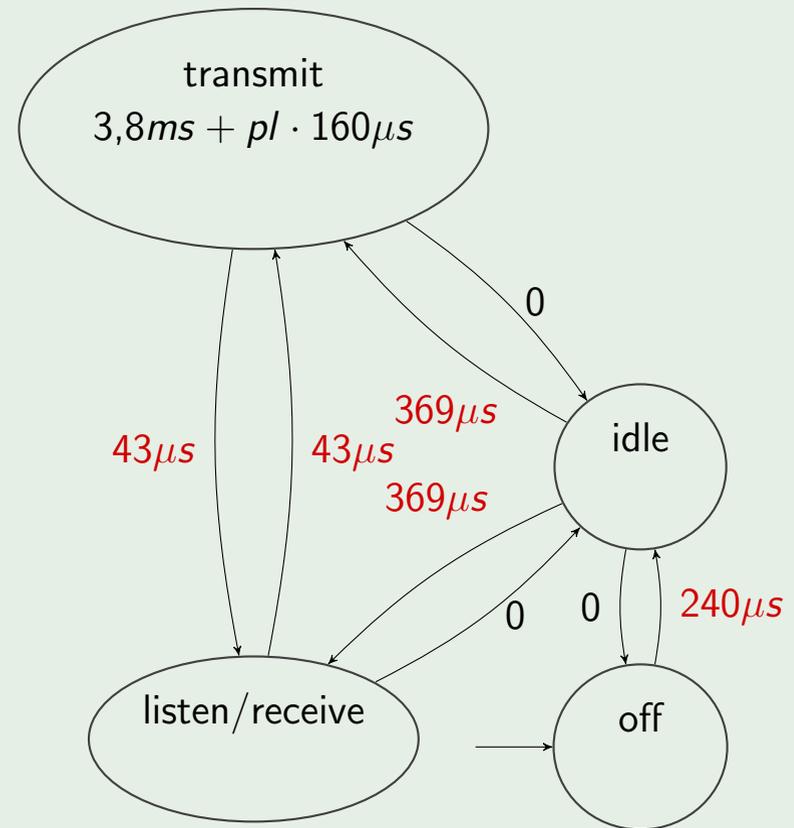
- Zeitangaben an Zuständen



Modellierung mit Zeitautomaten

- Zeitangaben an Zuständen
- ... Und an Transitionen

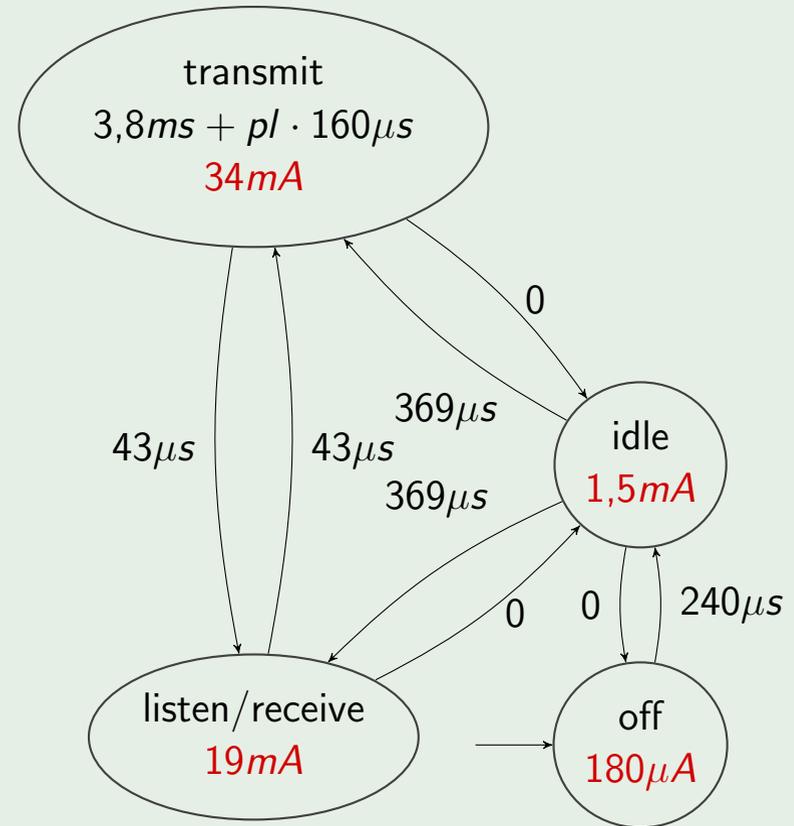
Beispiel (CC1200 @ 50kbps)



Modellierung mit Zeitautomaten

- Zeitangaben an Zuständen
- ... Und an Transitionen
- Kosten an Zuständen
 - Priced Timed Automaton (PTA)

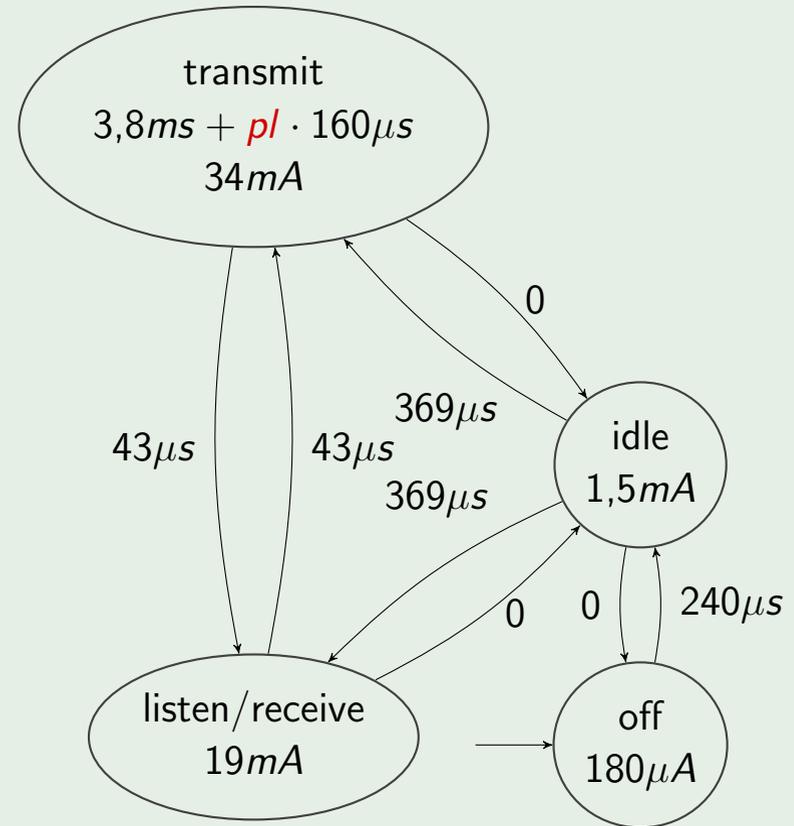
Beispiel (CC1200 @ 50kbps)



Modellierung mit Zeitautomaten

- Zeitangaben an Zuständen
- ... Und an Transitionen
- Kosten an Zuständen
 - Priced Timed Automaton (PTA)
- Parameter (Design/Laufzeit)

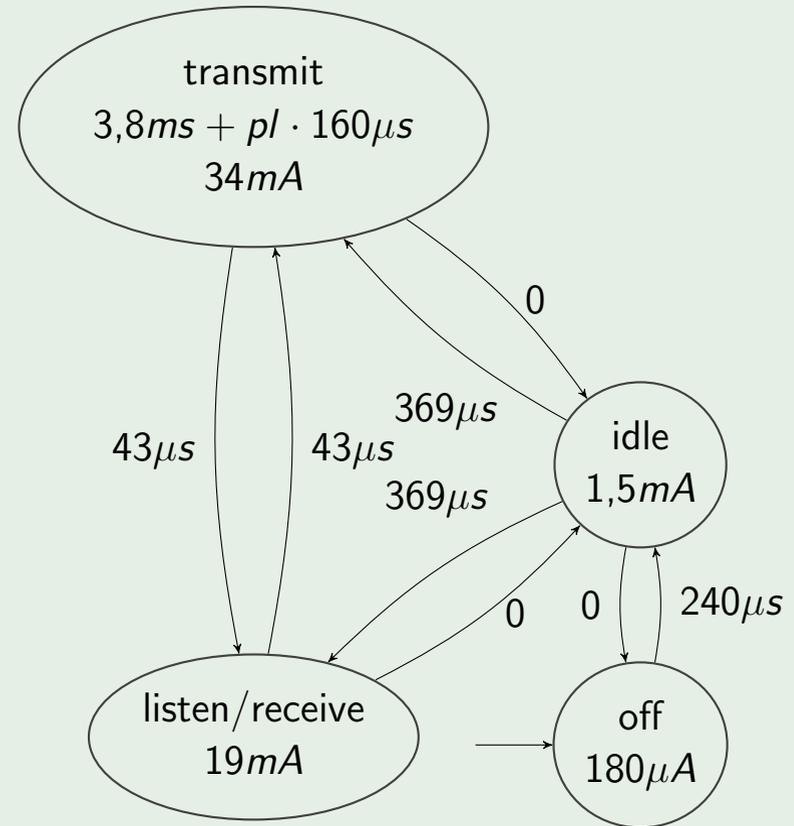
Beispiel (CC1200 @ 50kbps)



Modellierung mit Zeitautomaten

- Zeitangaben an Zuständen
- ... Und an Transitionen
- Kosten an Zuständen
 - Priced Timed Automaton (PTA)
- Parameter (Design/Laufzeit)
- Beachten: Variierende Definitionen in der Literatur
 - Meist gleich ausdrucksstark

Beispiel (CC1200 @ 50kbps)



Anwendung und Erweiterung von PTAs

- Sehr mächtiges und gut erforschtes Modell
- Unterstützt alle Online/Offline Modeling-Anforderungen

Anwendung und Erweiterung von PTAs

- Sehr mächtiges und gut erforschtes Modell
- Unterstützt alle Online/Offline Modeling-Anforderungen
 - Model Checking nur wenn ganzes System als PTA

Anwendung und Erweiterung von PTAs

- Sehr mächtiges und gut erforschtes Modell
- Unterstützt alle Online/Offline Modeling-Anforderungen
 - Model Checking nur wenn ganzes System als PTA
- Szenariobasierte Evaluation
- Energieverwaltung zur Laufzeit

Anwendung und Erweiterung von PTAs

- Sehr mächtiges und gut erforschtes Modell
- Unterstützt alle Online/Offline Modeling-Anforderungen
 - Model Checking nur wenn ganzes System als PTA
- Szenariobasierte Evaluation
- Energieverwaltung zur Laufzeit
- Probabilistische Erweiterung möglich:
Wahrscheinlichkeitsverteilungen für Zeit und Energie

- PTA im Treiber: Aufwändig und größtenteils redundant

- PTA im Treiber: Aufwändig und größtenteils redundant

```
int send(char* packet, int packet_length) { ... }
```

- PTA im Treiber: Aufwändig und größtenteils redundant

```
int send(char* packet, int packet_length) { ... }
```

```
energy_send(packet_length) = 38nAh + packet_length · 1,5nAh
```

Macro Modeling

- PTA im Treiber: Aufwändig und größtenteils redundant

```
int send(char* packet, int packet_length) { ... }
```

$\text{energy_send}(\text{packet_length}) = 38nAh + \text{packet_length} \cdot 1,5nAh$

send(pl, pwr)	8B	32B	128B	512B
+15 dBm
+14 dBm
+10 dBm

Macro Modeling

- PTA im Treiber: Aufwändig und größtenteils redundant

```
int send(char* packet, int packet_length) { ... }
```

$energy_send(packet_length) = 38nAh + packet_length \cdot 1,5nAh$

send(pl, pwr)	8B	32B	128B	512B
+15 dBm
+14 dBm
+10 dBm

- Energiemodell auf Funktionsebene

Macro Modeling: Synthese

- Berechnung aus Messwerten [Tan+02] [Hur+11]
 - Modell wird vorgegeben:
$$\text{send}(\textit{length}) = \textit{setup_energy} + \textit{length} \cdot \textit{transmit_energy}$$

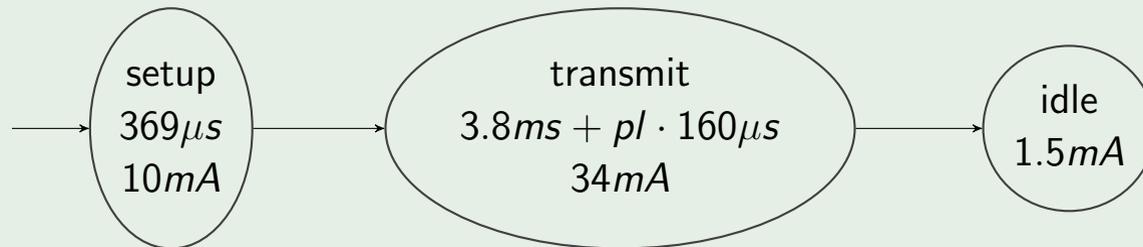
Macro Modeling: Synthese

- Berechnung aus Messwerten [Tan+02] [Hur+11]
 - Modell wird vorgegeben:
$$\text{send}(length) = \text{setup_energy} + length \cdot \text{transmit_energy}$$
 - Testprogramme für Treiberfunktionen und Parameter → Lineare Regression

Macro Modeling: Synthese

- Berechnung aus Messwerten [Tan+02] [Hur+11]
 - Modell wird vorgegeben:
 $\text{send}(\text{length}) = \text{setup_energy} + \text{length} \cdot \text{transmit_energy}$
 - Testprogramme für Treiberfunktionen und Parameter → Lineare Regression
- Oder aus vorhandenen PTAs [BGS12]
 - Kratos: PTA → Treiber-API und Accounting

Beispiel (Energieverbrauch für Radio.send)

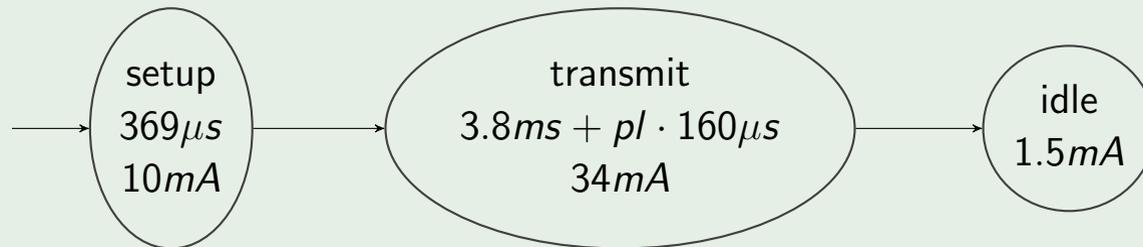


$$\text{energy_send}(32) = t_{\text{setup}} \cdot I_{\text{setup}} + t_{\text{transmit}} \cdot I_{\text{transmit}}$$

Macro Modeling: Synthese

- Berechnung aus Messwerten [Tan+02] [Hur+11]
 - Modell wird vorgegeben:
 $\text{send}(\text{length}) = \text{setup_energy} + \text{length} \cdot \text{transmit_energy}$
 - Testprogramme für Treiberfunktionen und Parameter → Lineare Regression
- Oder aus vorhandenen PTAs [BGS12]
 - Kratos: PTA → Treiber-API und Accounting

Beispiel (Energieverbrauch für Radio.send)

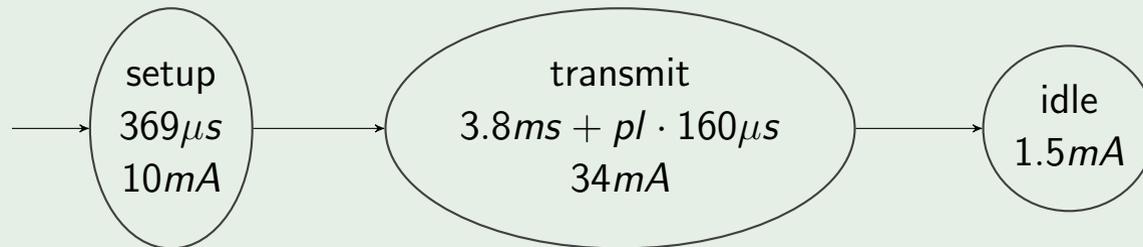


$$\text{energy_send}(32) = 369\mu\text{s} \cdot 10\text{mA} + (3,8\text{ms} + 5,12\text{ms}) \cdot 34\text{mA}$$

Macro Modeling: Synthese

- Berechnung aus Messwerten [Tan+02] [Hur+11]
 - Modell wird vorgegeben:
 $\text{send}(\text{length}) = \text{setup_energy} + \text{length} \cdot \text{transmit_energy}$
 - Testprogramme für Treiberfunktionen und Parameter → Lineare Regression
- Oder aus vorhandenen PTAs [BGS12]
 - Kratos: PTA → Treiber-API und Accounting

Beispiel (Energieverbrauch für Radio.send)

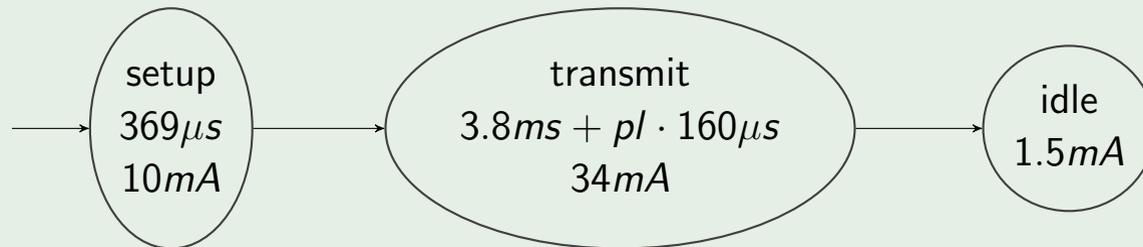


$$\text{energy_send}(32) = 85,3nAh$$

Macro Modeling: Synthese

- Berechnung aus Messwerten [Tan+02] [Hur+11]
 - Modell wird vorgegeben:
 $\text{send}(\text{length}) = \text{setup_energy} + \text{length} \cdot \text{transmit_energy}$
 - Testprogramme für Treiberfunktionen und Parameter → Lineare Regression
- Oder aus vorhandenen PTAs [BGS12]
 - Kratos: PTA → Treiber-API und Accounting

Beispiel (Energieverbrauch für Radio.send)



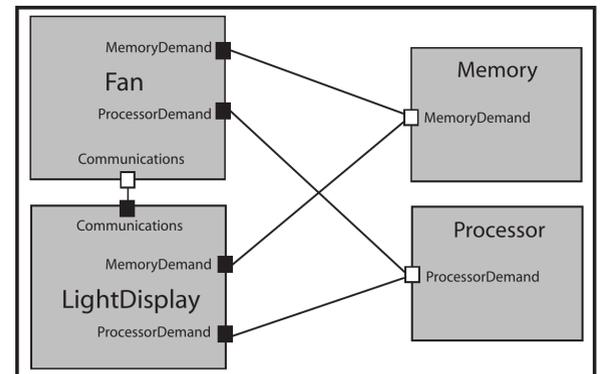
$$\text{energy_send}(32) = 85,3nAh$$

$$\text{energy_send}(pl) = 36,9nAh + pl \cdot 3,4nAh$$

- Prozessalgebren (ACSR) [LPS07] [VS08]

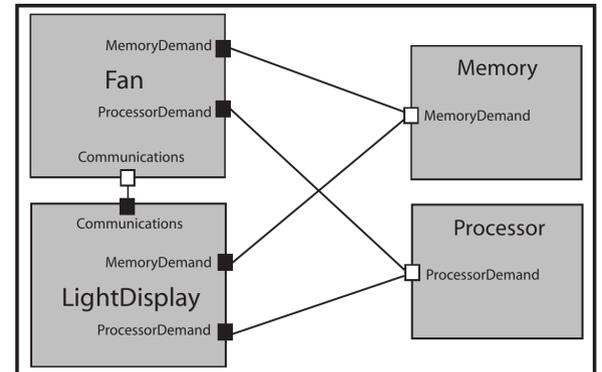
Weitere Modelle

- Prozessalgebren (ACSR) [LPS07] [VS08]
- UML/SPT [VS08]



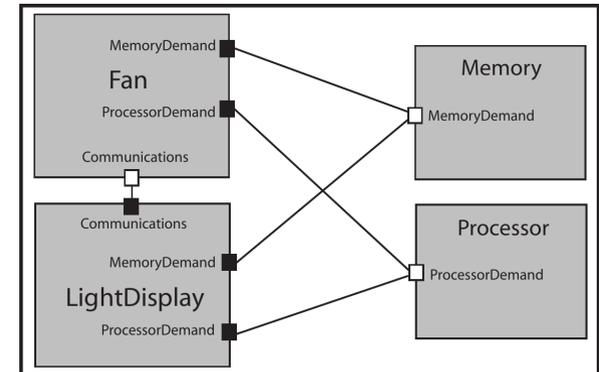
Weitere Modelle

- Prozessalgebren (ACSR) [LPS07] [VS08]
- UML/SPT [VS08]
- Timed Abstract State Machines



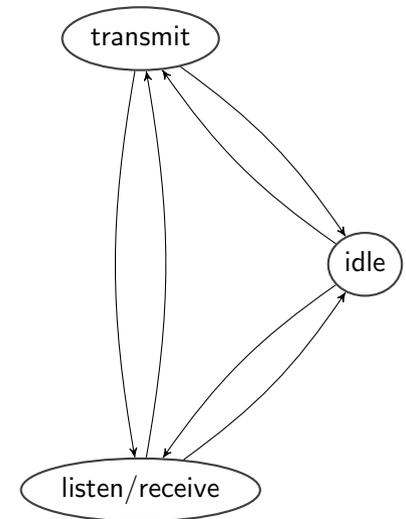
Weitere Modelle

- Prozessalgebren (ACSR) [LPS07] [VS08]
- UML/SPT [VS08]
- Timed Abstract State Machines
- Probabilistische Modelle
 - Z.B. auf Petrinetzbasis [Jun+06]



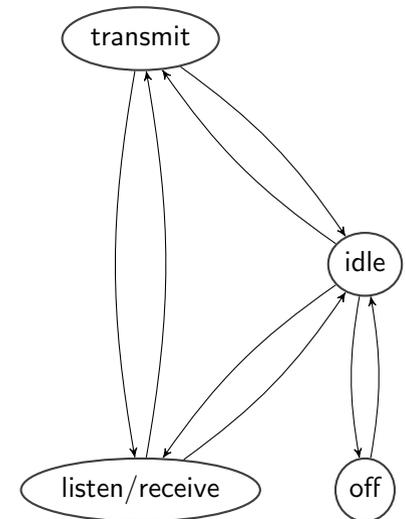
Modellierung von Funkchips

- Literatur: Three States Model



Modellierung von Funkchips

- Literatur: Three States Model



Modellierung von Funkchips

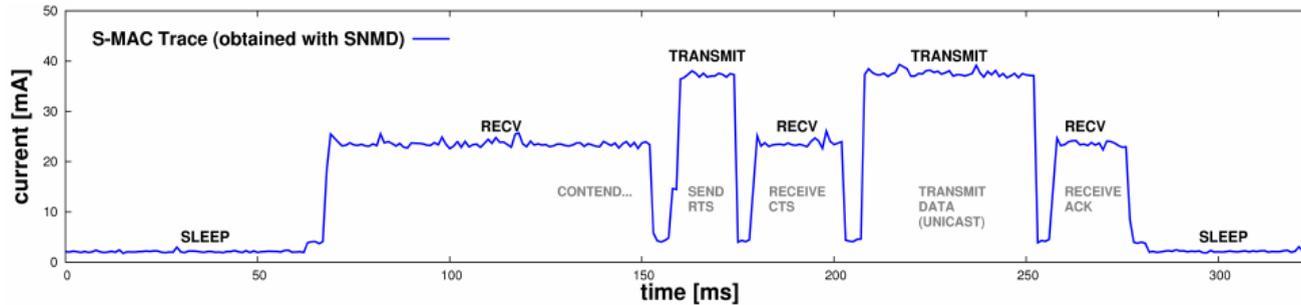
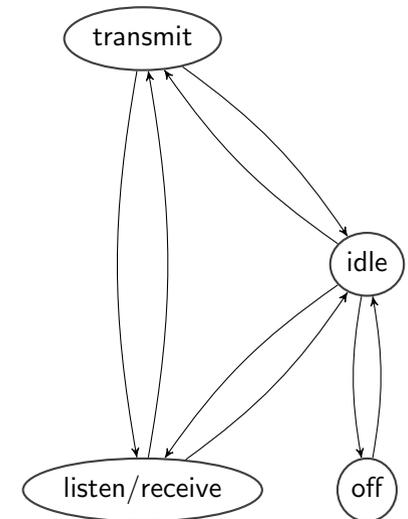


Abbildung: Stromverbrauch eines Funkchips aus [Hur+11]

- Literatur: Three States Model



Modellierung von Funkchips

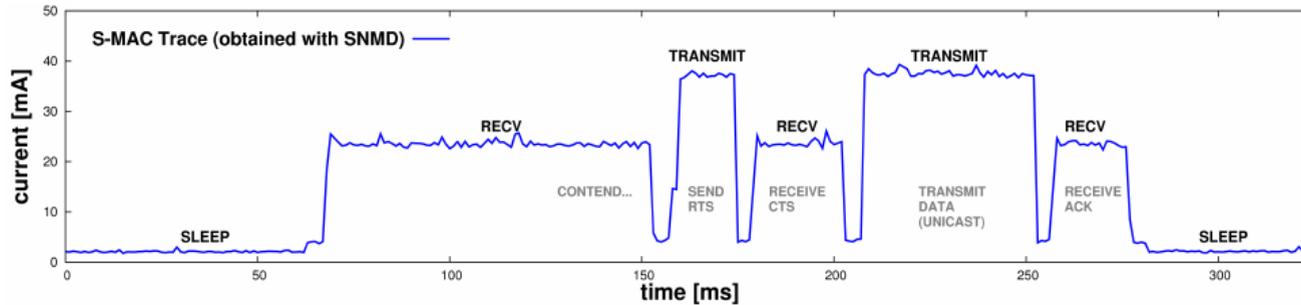
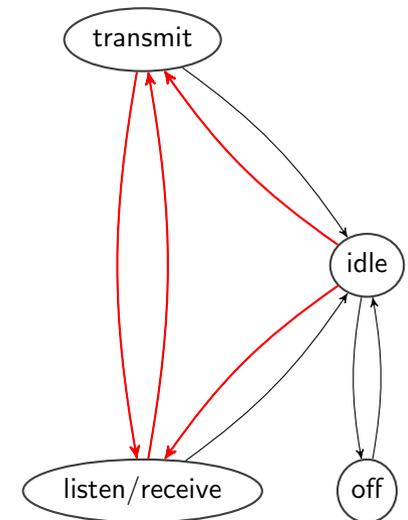


Abbildung: Stromverbrauch eines Funkchips aus [Hur+11]

- Literatur: Three States Model
- Implizite Wechselzustände beachten
 - Three States Model with State Transitions [Hur+11]



Modellierung von Funkchips

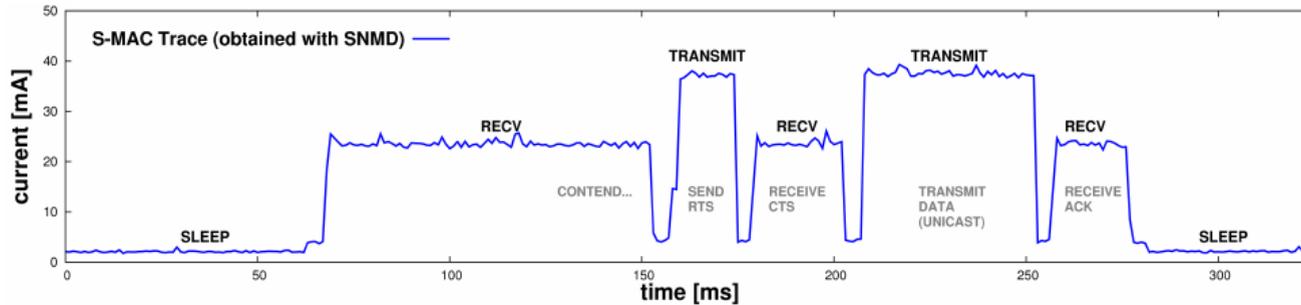
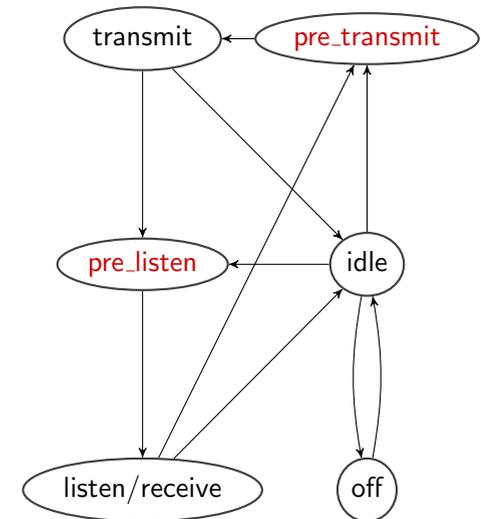


Abbildung: Stromverbrauch eines Funkchips aus [Hur+11]

- Literatur: Three States Model
- Implizite Wechselzustände beachten
 - Three States Model with State Transitions [Hur+11]
- Ggf. intuitiver: Explizite Zustände



Fehlerquellen und Herausforderungen

- Auch bei perfekten Modellen: Bis zu 5% Berechnungsfehler

Fehlerquellen und Herausforderungen

- Auch bei perfekten Modellen: Bis zu 5% Berechnungsfehler
 - Hardwaretoleranzen → Individuelle Modellkalibrierung

Fehlerquellen und Herausforderungen

- Auch bei perfekten Modellen: Bis zu 5% Berechnungsfehler
 - Hardwaretoleranzen → Individuelle Modellkalibrierung
- Energieberechnung verbraucht selbst Energie

Fehlerquellen und Herausforderungen

- Auch bei perfekten Modellen: Bis zu 5% Berechnungsfehler
 - Hardwaretoleranzen → Individuelle Modellkalibrierung
- Energieberechnung verbraucht selbst Energie
- Literatur: Oft RAM (wiederverwendbar) statt Energie (monoton)

Fehlerquellen und Herausforderungen

- Auch bei perfekten Modellen: Bis zu 5% Berechnungsfehler
 - Hardwaretoleranzen → Individuelle Modellkalibrierung
- Energieberechnung verbraucht selbst Energie
- Literatur: Oft RAM (wiederverwendbar) statt Energie (monoton)
- Hier: Energie weder wiederverwendbar noch monoton
 - Tagsüber wird Akku durch Solarzelle aufgeladen

Fehlerquellen und Herausforderungen

- Auch bei perfekten Modellen: Bis zu 5% Berechnungsfehler
 - Hardwaretoleranzen → Individuelle Modellkalibrierung
- Energieberechnung verbraucht selbst Energie
- Literatur: Oft RAM (wiederverwendbar) statt Energie (monoton)
- Hier: Energie weder wiederverwendbar noch monoton
 - Tagsüber wird Akku durch Solarzelle aufgeladen
- Detailgrad der Modelle

Zusammenfassung

- Es gibt viele teils sehr komplexe Modelle

Zusammenfassung

- Es gibt viele teils sehr komplexe Modelle
- Literatur betrachtet oft andere Ressourcen als Energie

Zusammenfassung

- Es gibt viele teils sehr komplexe Modelle
- Literatur betrachtet oft andere Ressourcen als Energie
- Modell: Möglichst detailliert
- Messwerte: Möglichst viele

Zusammenfassung

- Es gibt viele teils sehr komplexe Modelle
- Literatur betrachtet oft andere Ressourcen als Energie
- Modell: Möglichst detailliert
- Messwerte: Möglichst viele

Vorschlag:

- PTAs mit expliziten Wartezuständen

Zusammenfassung

- Es gibt viele teils sehr komplexe Modelle
- Literatur betrachtet oft andere Ressourcen als Energie
- Modell: Möglichst detailliert
- Messwerte: Möglichst viele

Vorschlag:

- PTAs mit expliziten Wartezuständen
- Energievorhersage: Macro Models auf PTA-Basis

Zusammenfassung

- Es gibt viele teils sehr komplexe Modelle
- Literatur betrachtet oft andere Ressourcen als Energie
- Modell: Möglichst detailliert
- Messwerte: Möglichst viele

Vorschlag:

- PTAs mit expliziten Wartezuständen
- Energievorhersage: Macro Models auf PTA-Basis
- Falls möglich: Individuelle Modellkalibrierung

Zusammenfassung

- Es gibt viele teils sehr komplexe Modelle
- Literatur betrachtet oft andere Ressourcen als Energie
- Modell: Möglichst detailliert
- Messwerte: Möglichst viele

Vorschlag:

- PTAs mit expliziten Wartezuständen
- Energievorhersage: Macro Models auf PTA-Basis
- Falls möglich: Individuelle Modellkalibrierung
- Falls Interesse: Probabilistische PTAs

- [BGS12] Markus Buschhoff, Christian Günter und Olaf Spinczyk. “A unified approach for online and offline estimation of sensor platform energy consumption”. In: Proceedings of the 8th International Wireless Communications and Mobile Computing Conference (IWCMC '12). Limassol, Cyprus, Aug. 2012, S. 1154–1158. DOI: 10.1109/IWCMC.2012.6314369.
- [DMC03] Merijn De Jonge, Johan Muskens und Michel Chaudron. “Scenario-based prediction of run-time resource consumption in component-based software systems”. In: Proceedings: 6th ICSE Workshop on Component Based Software Engineering: Automated Reasoning and Prediction. 2003.

- [Hur+11] Philipp Hurni u. a. “On the Accuracy of Software-Based Energy Estimation Techniques”. In: *Wireless Sensor Networks*. Bd. 6567. *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2011, S. 49–64. DOI: 10.1007/978-3-642-19186-2_4.
- [Jun+06] MeuseN.O. Junior u. a. “Analyzing Software Performance and Energy Consumption of Embedded Systems by Probabilistic Modeling: An Approach Based on Coloured Petri Nets”. In: *Petri Nets and Other Models of Concurrency - ICATPN 2006*. Bd. 4024. *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2006, S. 261–281. DOI: 10.1007/11767589_15.

- [Kwi07] Marta Kwiatkowska. “Quantitative verification: models techniques and tools”. In: Proceedings of the the 6th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on The foundations of software engineering. ACM. 2007, S. 449–458.
- [LPS07] Insup Lee, Anna Philippou und Oleg Sokolsky. “Resources in process algebra”. In: The Journal of Logic and Algebraic Programming 72.1 (2007), S. 98–122.
- [Tan+02] Tat Kee Tan u. a. “High-level energy macromodeling of embedded software”. In: IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 21.9 (2002), S. 1037–1050.

- [TRJ02] Tat Kee Tan, Anand Raghunathan und Niraj K Jha. “Embedded operating system energy analysis and macro-modeling”. In: *Computer Design: VLSI in Computers and Processors*, 2002. Proceedings. 2002 IEEE International Conference on. IEEE. 2002, S. 515–522.
- [VS08] Aneta Vulgarakis und Cristina Seceleanu. “Embedded systems resources: Views on modeling and analysis”. In: *Computer Software and Applications*, 2008. COMPSAC’08. 32nd Annual IEEE International. IEEE. 2008, S. 1321–1328.