

ITS-E

Explorativer Roboter

Jonas Heberer, Rene Bamberger, Jan Reinhard

Hochschule Rhein Main, Fachbereich Design Informatik Medien,
Deutschland, Unter den Eichen 5, 65195 Wiesbaden

Inhaltsverzeichnis

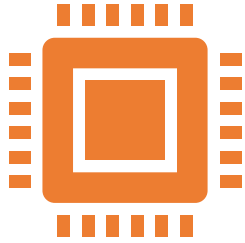
- Einführung
- Grob Design
- Modulbeschreibung
- Aufgetretene Probleme und Änderungen
- Fazit
- Praktische Präsentation

Einführung

Idee: Autarker Roboter

- Aus nachhaltigen Materialien
- Aufladen des Akkus mittels Solarpanel
- Logging anfallender Daten mittels BLE
- Kartenbau und Lokalisierung

Einführung



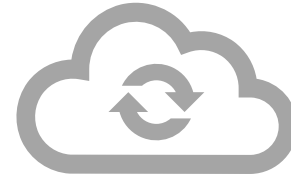
Hardware

Adafruit Feather

Ultraschall-Sensor US-100

Motortreiber Adafruit DRV8833

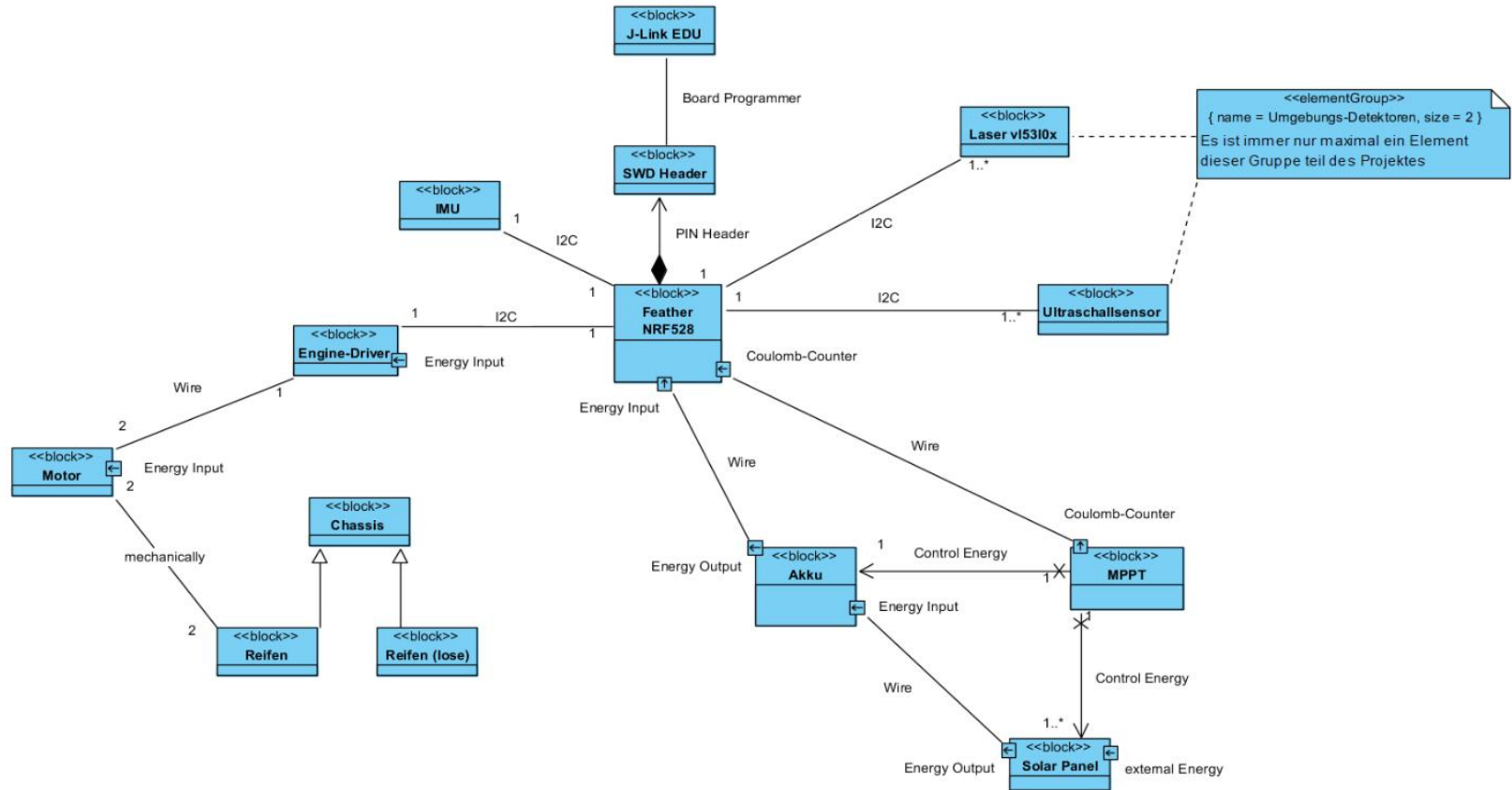
Motoren Motraxx GM12F-N20VS



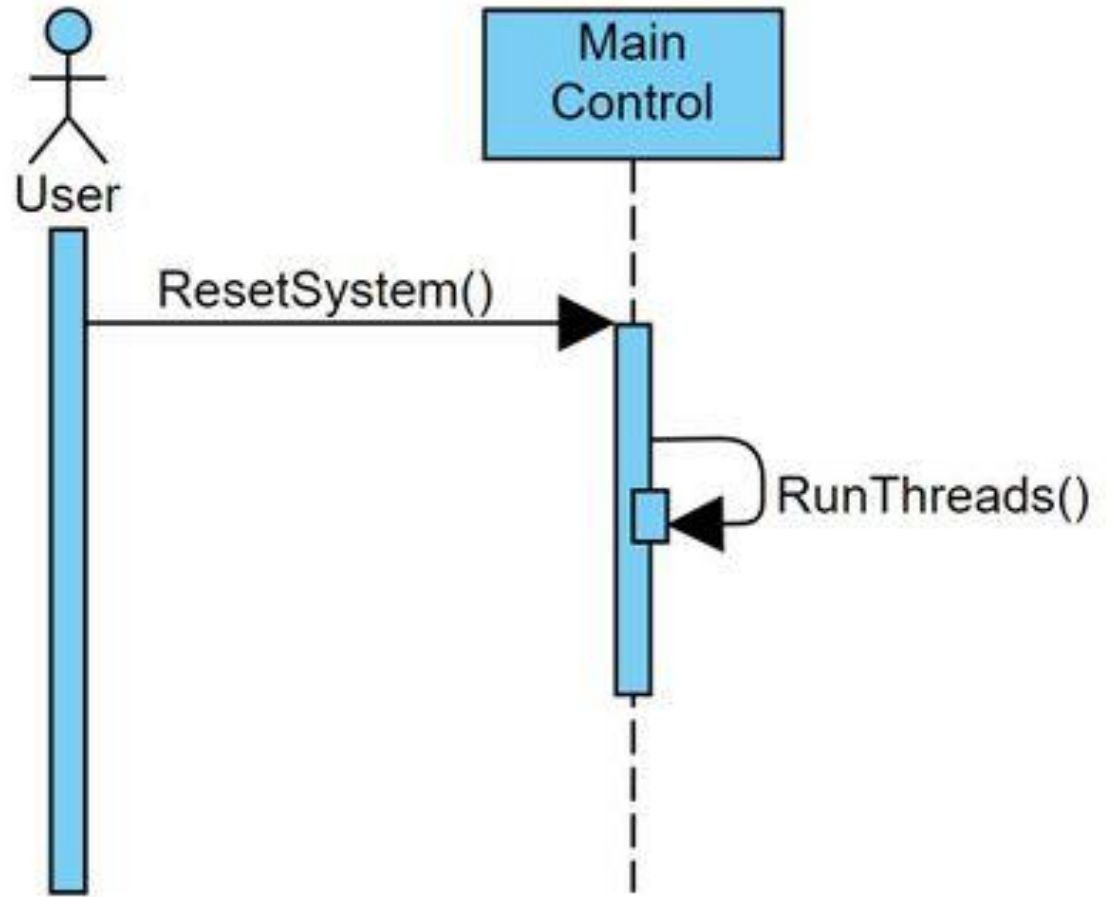
Software

Zephyr RTOS

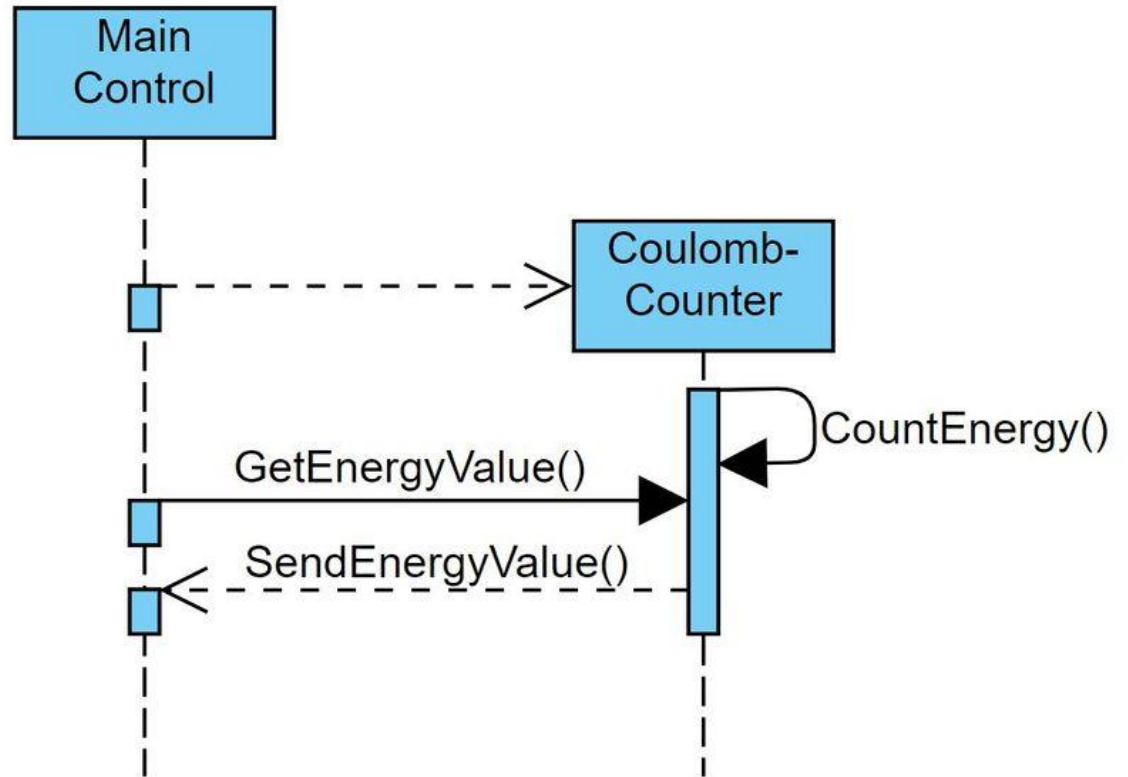
Grob Design



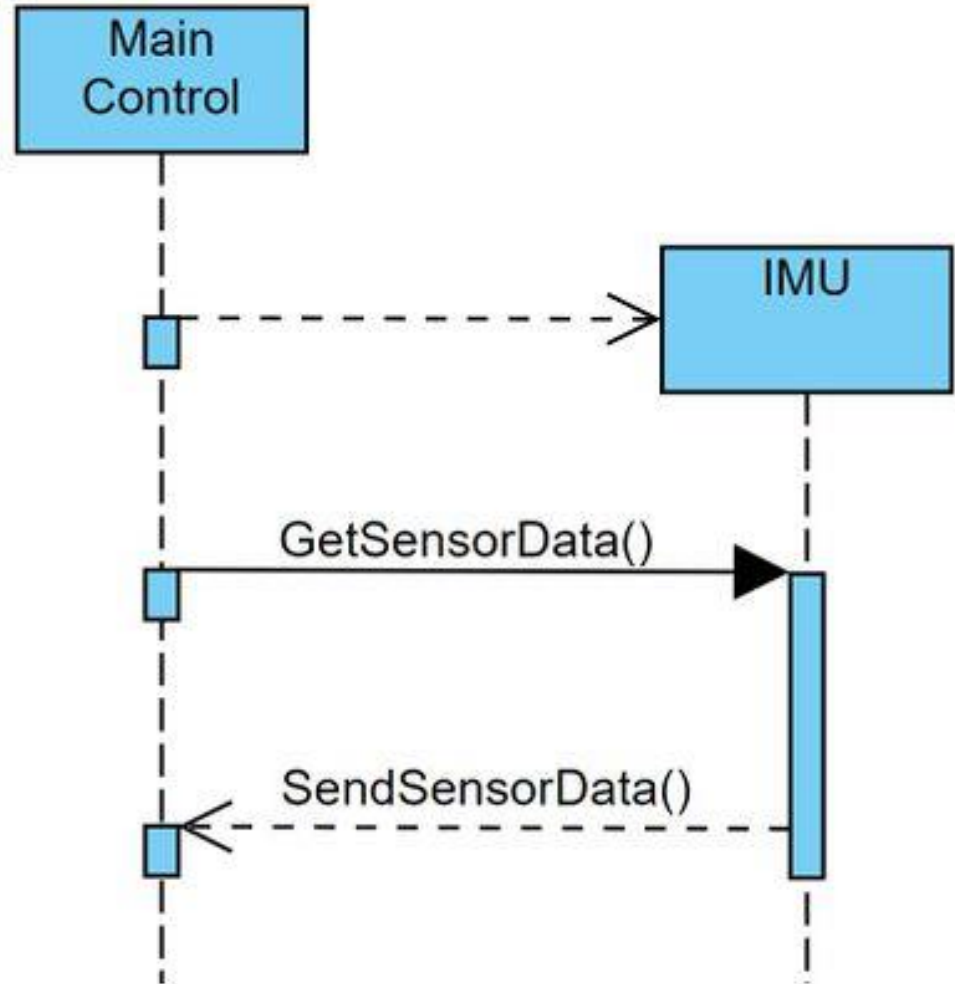
Grob Design



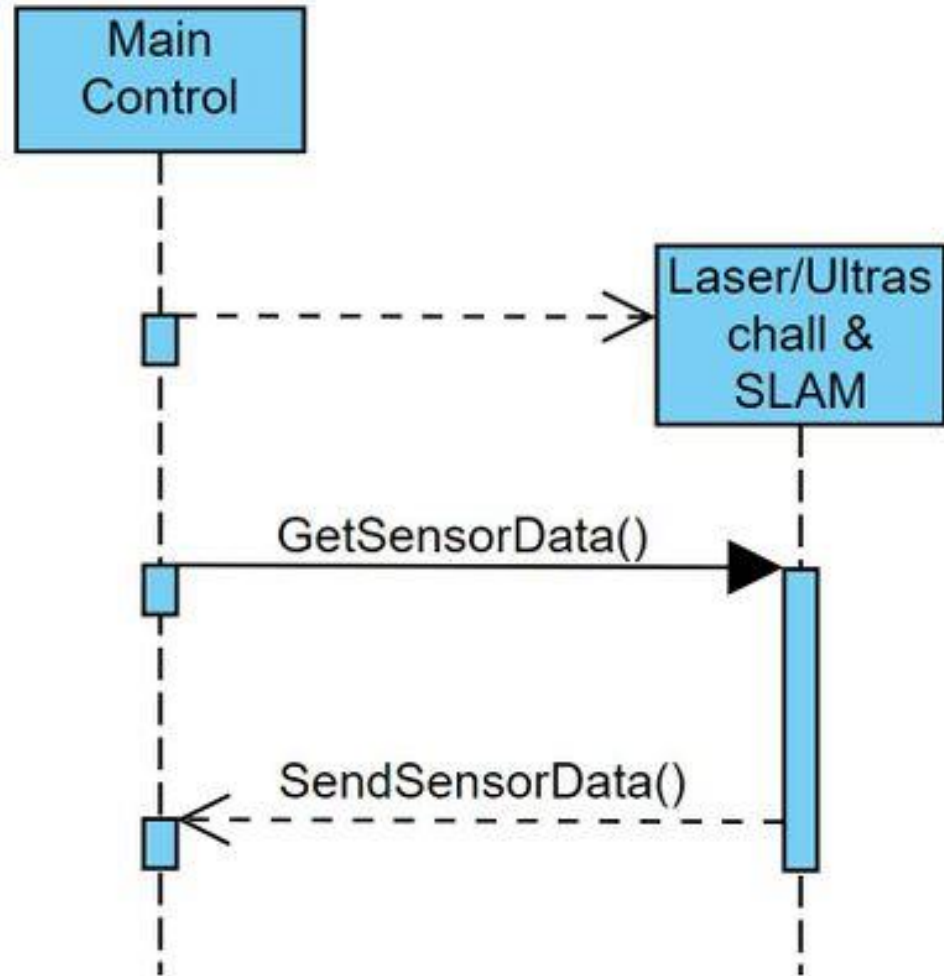
Grob Design



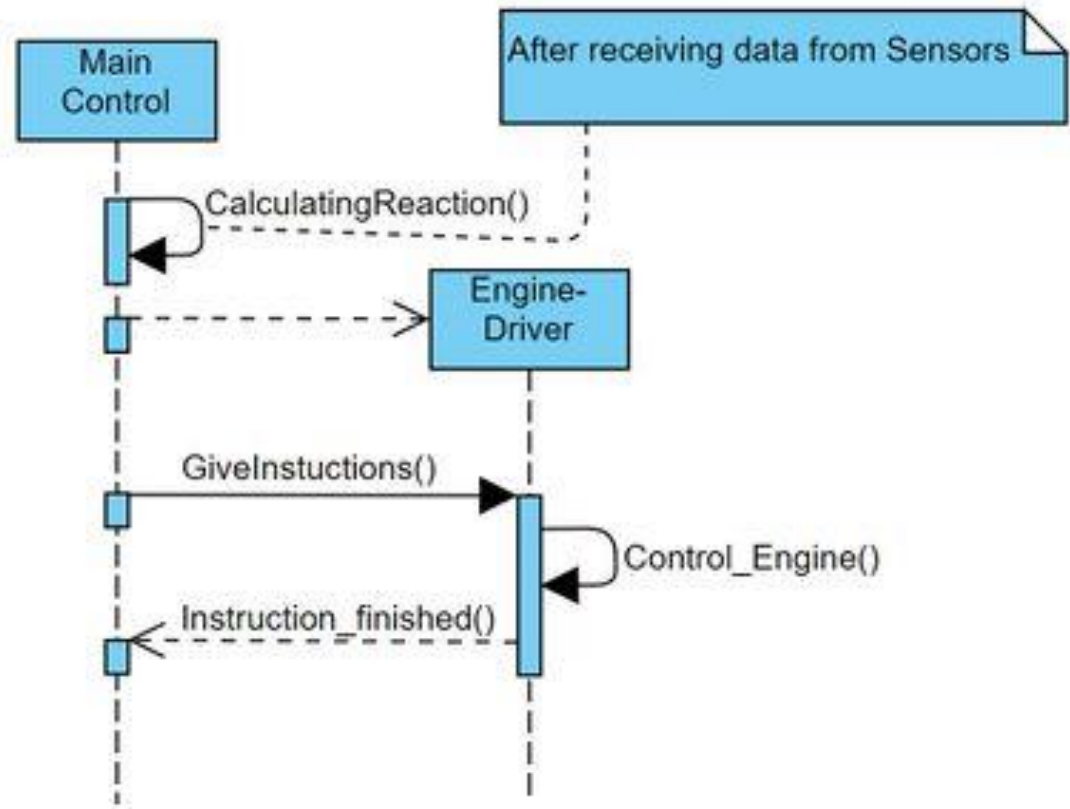
Grob Design



Grob Design

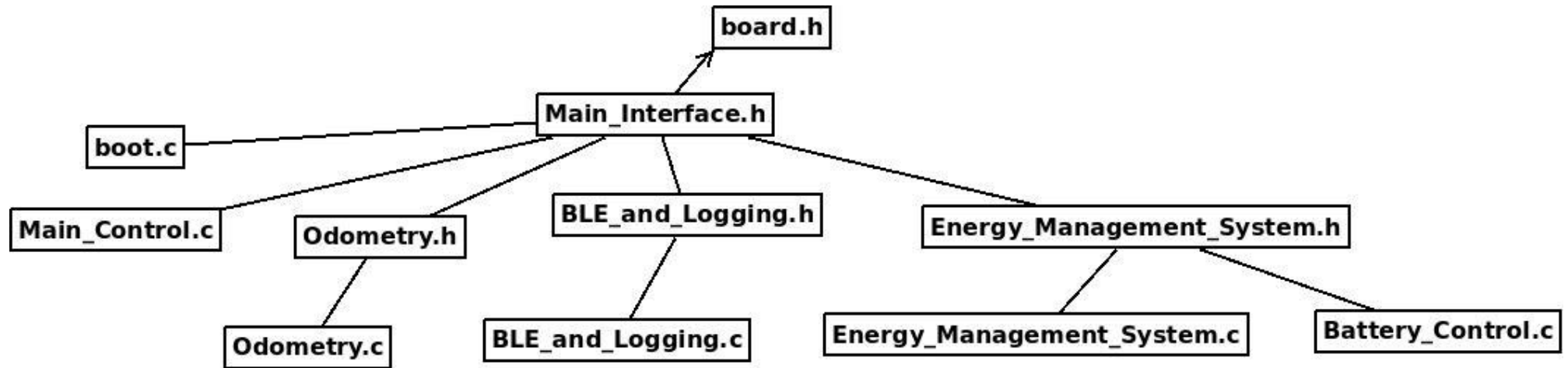


Grob Design



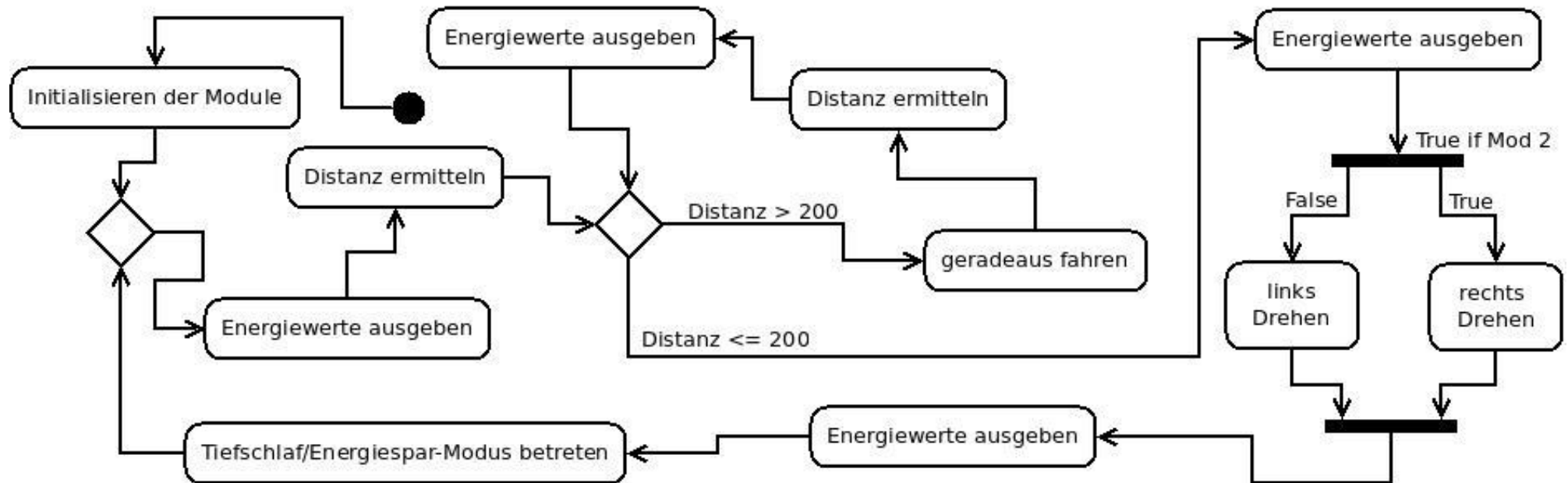
Modulbeschreibung

Dateistruktur



Modulbeschreibung

Main Control



Modulbeschreibung BLE und Logging

- Eigenschaften:
 - sendet Informationen via BLE
 - Datum, Uhrzeit, Art des Moduls, Return-Wert und weitere Informationen werden versendet
 - zum Empfang und weiteren Verarbeitung der Daten wird ein BLE-USB-Stick und ein Python-Script verwendet

Funktions-Deklarationen des BLE-Moduls:

```
extern void set_msg(int pos, unsigned char hex);  
extern void bt_ready(int err);  
extern void ble_and_logging(void* arg1, void* arg2, void* arg3);  
extern void ble_set_data(char *data, int art_der_daten);
```

Modulbeschreibung Energy Management System

- Eigenschaften:
 - Gibt auf Anfrage Energie-Werte zurück
 - Batterie-Stand
 - Verbrauch
 - Einnahmen
 - Erfasst Energie-Werte je nach momentanem Zustand des Systems
 - Startet Energiespar-Modus

Energiespar-Modus:

```
sys_set_power_state(SYS_POWER_STATE_SLEEP);  
  
k_sched_lock();  
  
k_sleep( (SLEEP_SEC*60) * arg);  
  
k_sched_unlock();
```

Modulbeschreibung Odometrie

- Eigenschaften:
 - Ansteuern und Auslesen des Ultraschallsensors

Deklaration der Haupt-Funktionen:

```
int32_t odm_sensor_init();  
int32_t odm_sensor_value_get(odm_sensor_value_t*);  
  
int32_t odm_sonic_init();  
int32_t odm_sonic_value_get(odm_sensor_value_t*);  
  
int32_t odm_laser_init();  
int32_t odm_laser_value_get(odm_sensor_value_t*);
```

Aufgetretene Probleme und Änderungen

- PWM Driver funktioniert nicht
 - Alternativer PWM-Driver gefunden
 - Ist nicht für mehrere Pins ausgelegt
- I2C macht auf NRF-Devices Probleme
 - Laser-Sensor
 - Laderegler MPPT
 - IMU
- Extended Kalman Filter
 - vollständige Implementierung würde zeitlichen Rahmen sprengen
- Zeitverlust durch Materialbeschaffung
- Energie-Pegel nicht auslesbar
 - Keine realistischen Werte bekommen
- Motor-Treiber defekt
 - FLT (Fault) Pin zeigte internen Fehler / Pin war nicht mehr ansprechbar

Praktische Präsentation

Versuchsbeschreibung:

- **Hauptaufgabe**
 - Schrittweises Fahren bis 20 cm von Gegenstand entfernt
 - Halte an
 - Drehen um 180° (zwei mal 90)
 - Energiesparmodus aktivieren
- **Parallele Aufgaben**
 - Energie messen
 - Daten über BLE senden

Fazit

- Es konnte eine (hoffentlich) lauffähige Demo erstellt werden
- Durch die Bearbeitung der jeweiligen Module konnten wertvolle Erkenntnisse gesammelt werden
- Das Projekt konnte so gestaltet werden, dass es in verschiedene Richtungen ausbaufähig ist