

# Smart Window

---

Vorstellung des Projektes „Smart Window“

Mailina Lohmann, Patrick M\"olk,  
Sebastian Kaestner, Sebastian Heinrich, Simon  
Stallbaum

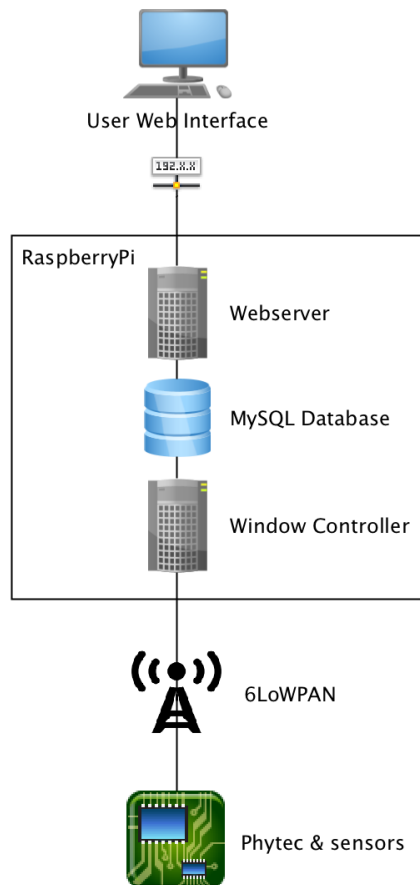
# Ablauf

Smart  
Window

- Gesamtkonzeptes
- Einzelfunktionen:
  - Microcontroller, Sensoren, Aktoren
  - Kommunikation
  - Konzept, Steuerung
  - Datenbank
  - Webserver
- Live Vorführung
- Fazit
- Feedback

# Gesamtkonzept

Smart  
Window

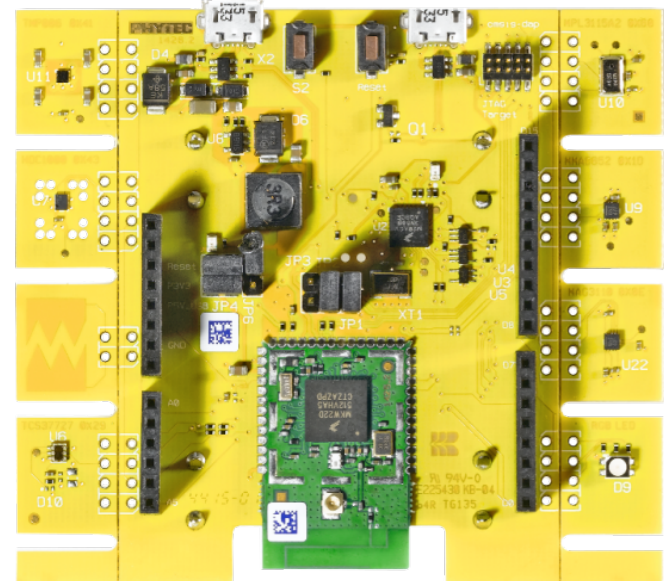


- Verbesserung der Umgebungseinflüsse
- Softwarearchitekturen
- Hardware

# Mikrocontroller

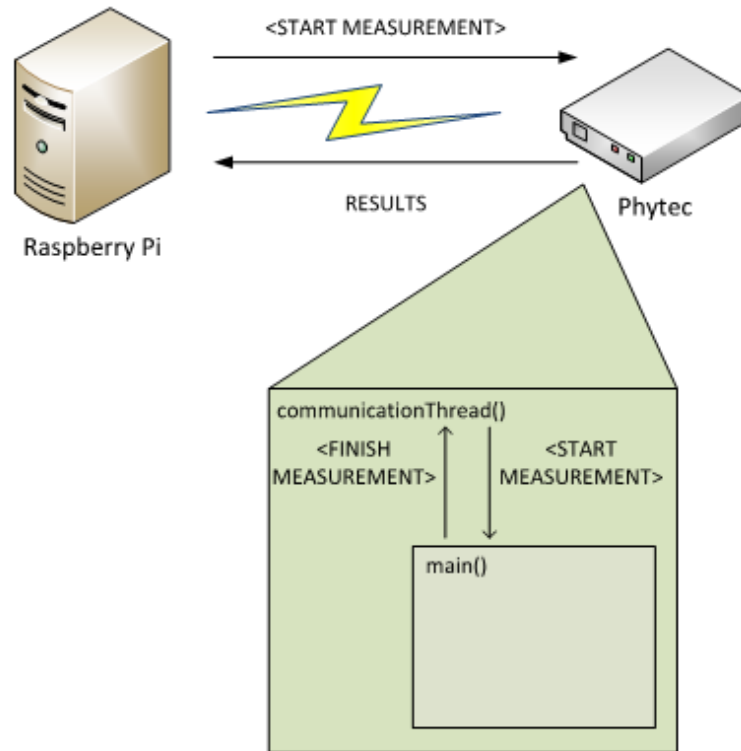
Smart  
Window

- Phytec-Board
- Programmentwicklung mit RIOT
- 2 Threads :
  - Thread 1:  
Auslesen + Aufbereiten der Messwerte, Aktorsteuerung
  - Thread 2:  
Kommunikation



# Mikrocontroller Funktion

Smart  
Window



- Eigenes einfaches Kommunikationsprotokoll

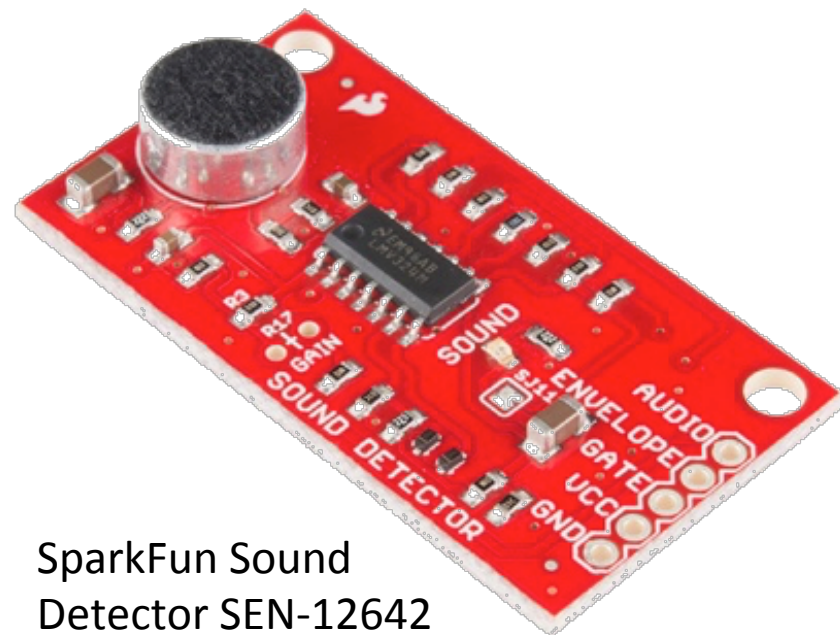
# Sensoren

- Schalldruckpegel:

- $U_{\text{mic}} \sim p$

$$\tilde{p} = \sqrt{\frac{1}{n} \sum_{i=1}^n (u_i \cdot k)^2}$$

$$k = 0,02457 \frac{\text{Pa}}{\text{V}}$$



SparkFun Sound  
Detector SEN-12642

- Berechnung des Schalldruckpegels

$$L_p = 20 \cdot \log \left( \frac{\tilde{p}}{p_0} \right)$$

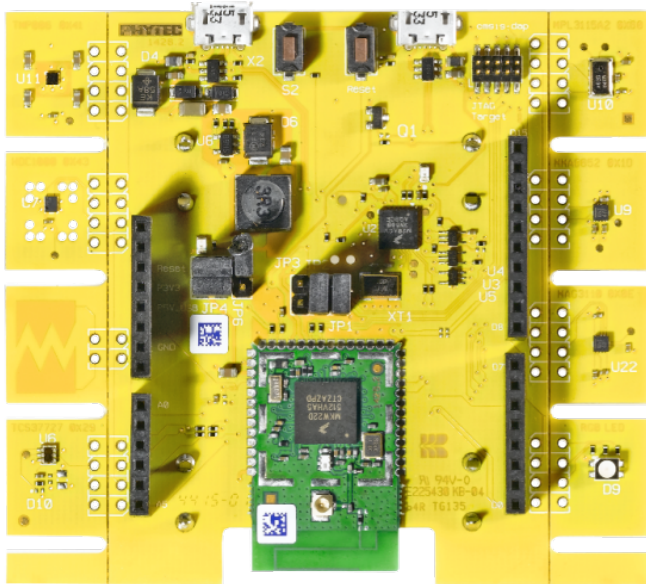
$$p_0 = 20 \mu Pa$$

- $p_0$  Schalldruck für menschliches Gehör bei 1kHz

# Sensoren

Smart  
Window

- Von Phyttec:
  - Temperatur
  - Luftfeuchtigkeit



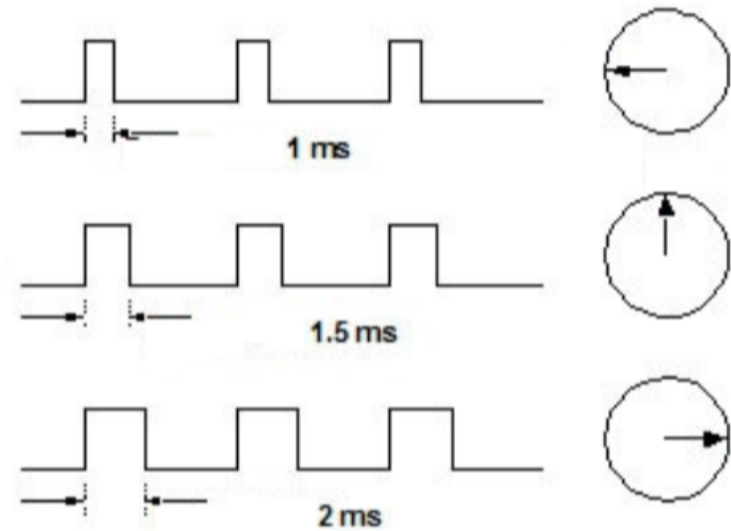
- MQ-135 Gas Sensor:
  - Luftqualität



# Aktor

Smart  
Window

- Servo Motor:
  - PWM-Ansteuerung



# Kommunikation

Smart  
Window

- 6LoWPAN
  - Kernel, Channel, Pan\_Id
  - UDP Header Compression

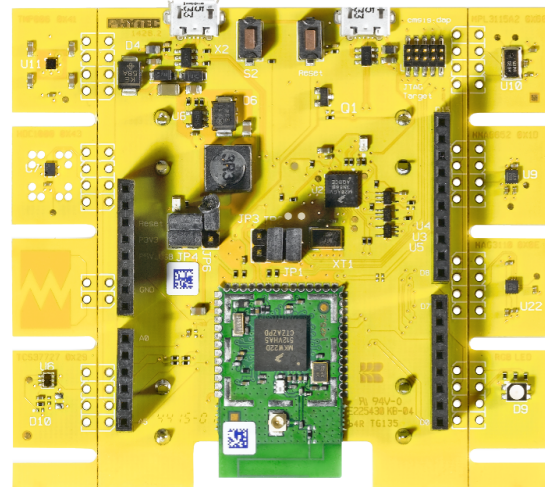


# Kommunikation

- Funktionen für RIOT

```
int sw_network_init(void); // set channel, pan_id
```

```
int sw_send(char *dst_addr, char *data);
```



# Kommunikation Kommandos

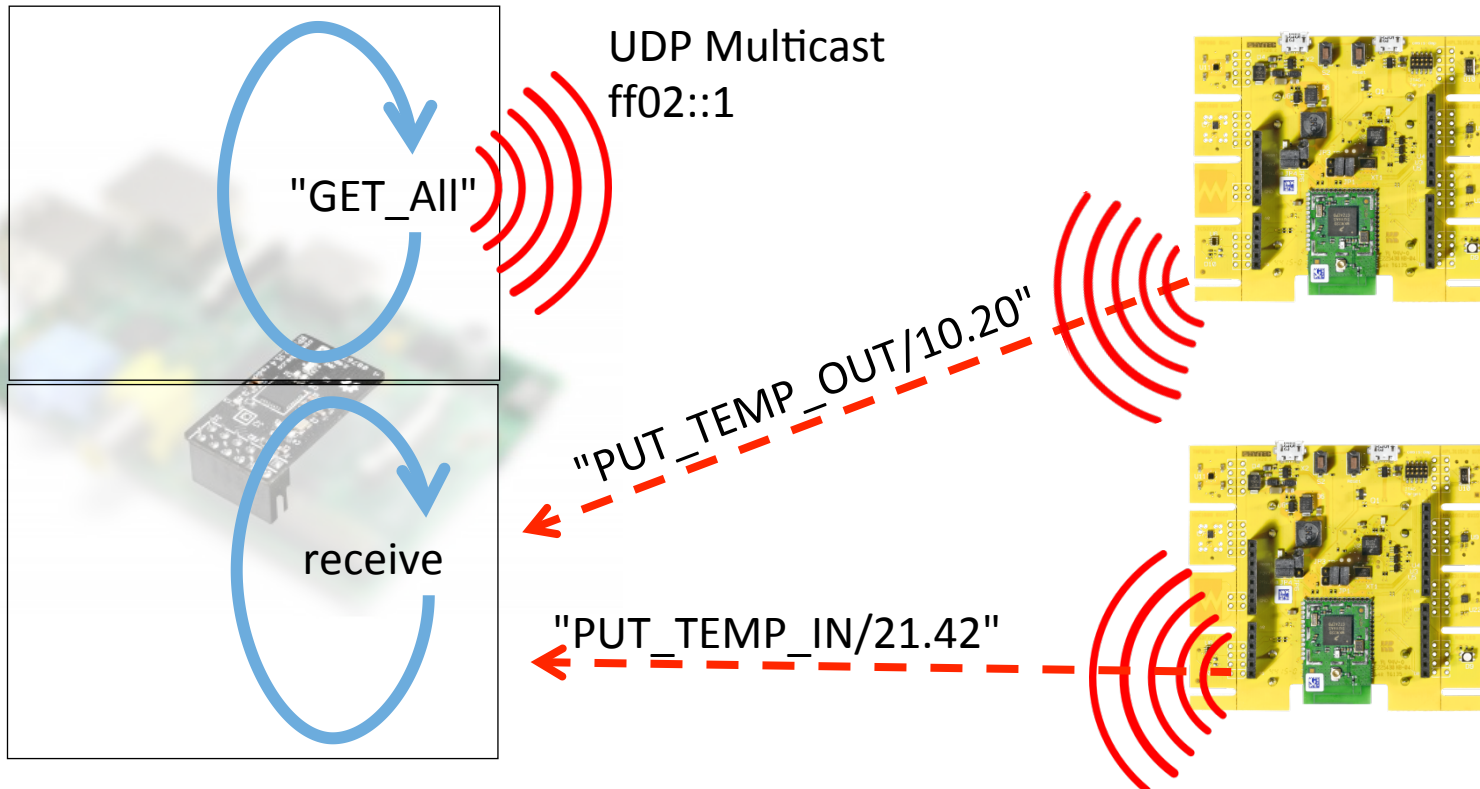
Smart  
Window

- String basierte Kommandos
- "<cmd>\_<sensor>\_<position>/<value>"
  - cmd: "GET", "PUT"
  - sensor: "TEMP", "VOLUME", etc.
  - position: "IN", "OUT"
  - Sonderkommando: "GET\_All"
- Feature Request:
  - COAP

# Kommunikation, Ablauf

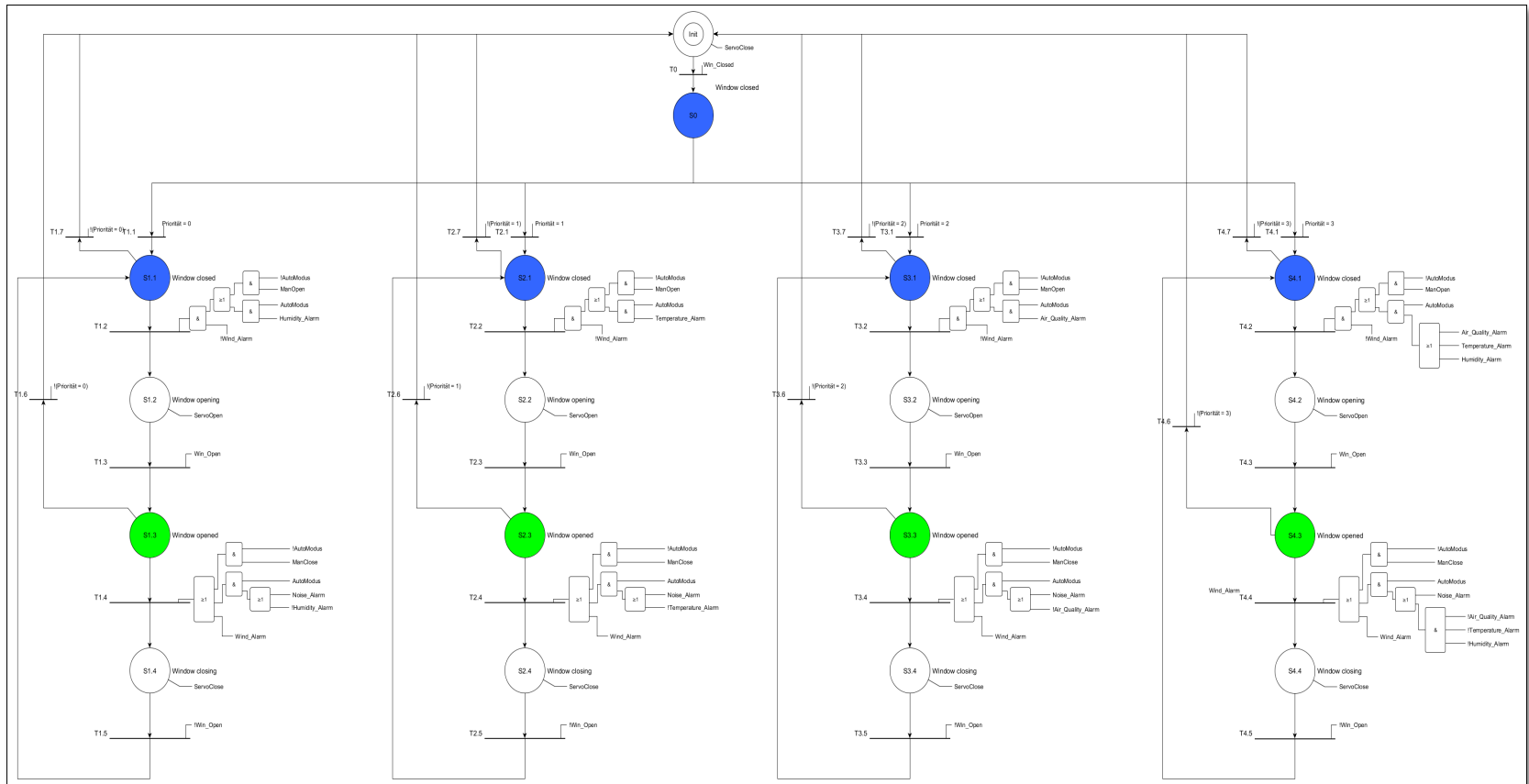
Smart  
Window

- Zwei Endlosschleifen (Threads)



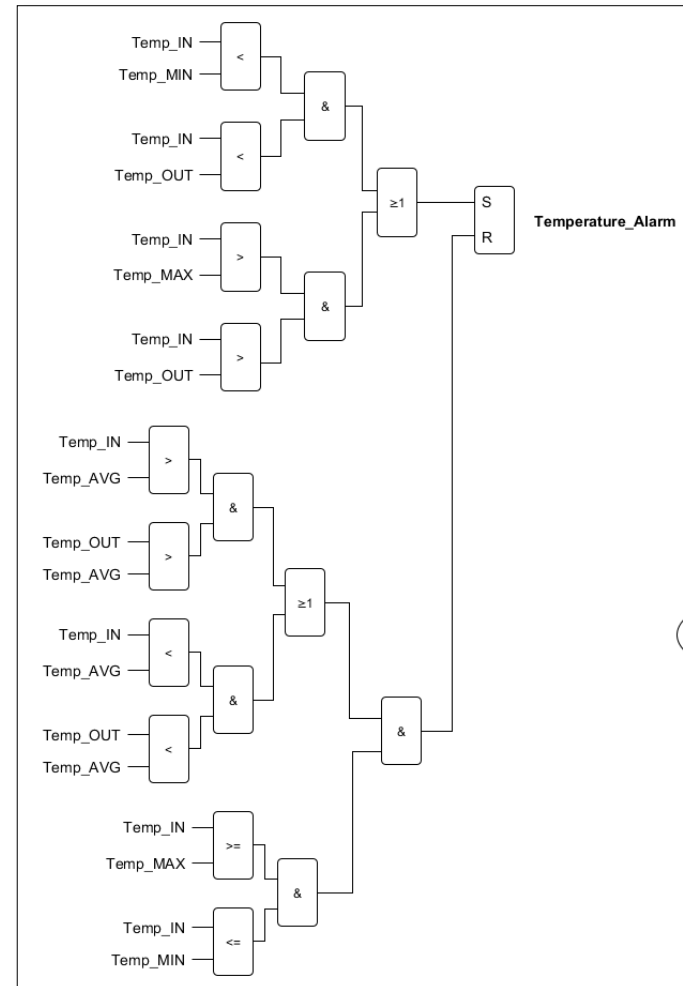
# Konzeptentwurf

# Smart Window



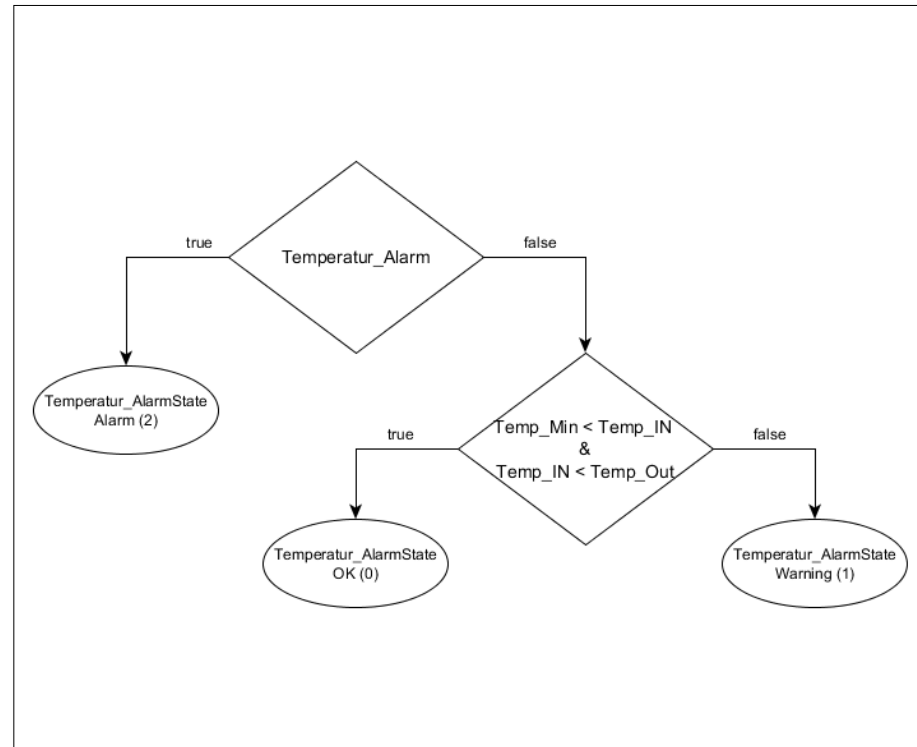
# Grenzwertberechnung

- Alarm, wenn:
    - Innenwert außerhalb Komfortbereich
  - UND
  - Außenwert besser als Innenwert
- Alarm, erlischt:
    - Innenwert im Komfortbereich
  - UND
  - Innenwert und Außenwert selbe Ausrichtung zum mittleren Komfortwert



# Alarmstatus

- Status 2
  - Alarm
- Status 1
  - Warnung
- Status 0
  - Alles OK





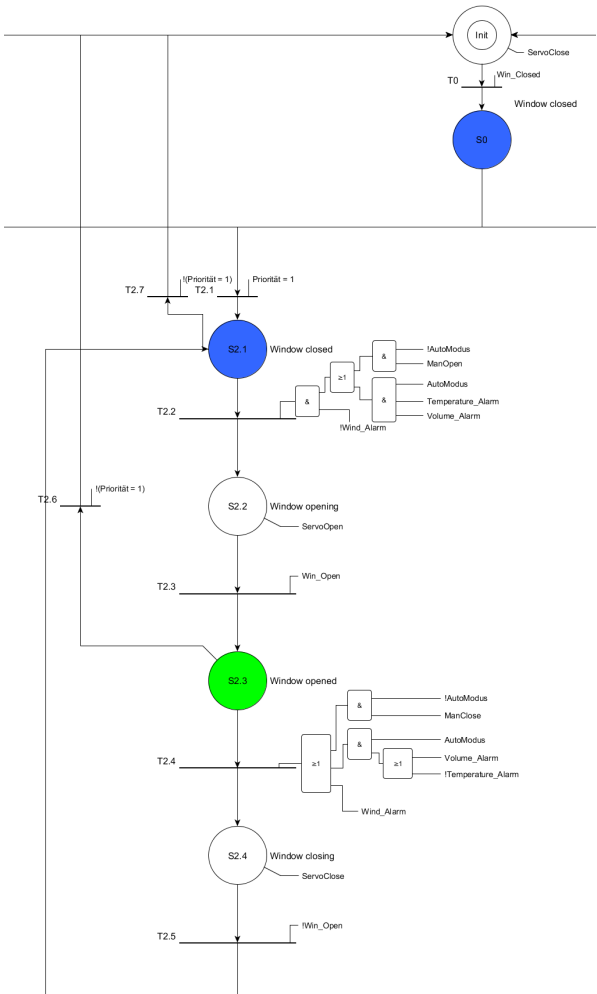
# Steuerung

- Moore Automat
- Ablauf in Pseudo-Code:

```
// setzen des Anfangszustands
while(1){
    // 1. lesen der Daten aus der DB
    // 2. Überprüfen der Alarme
    // 3. schreiben von Alarm Zuständen in DB
    // 4. Codierung des Petrinetzes
    // 5. Zuweisung von Ausgängen
}
```

# Steuerung

# Smart Window



```

// Anlegen die Variablen
bool init = true; // set initialisation state
bool state0 = false;
bool state11 = false;
bool state12 = false;
bool state13 = false;
bool state14 = false;
...

while(1){
    // read Auto/Manual out of database
    // read current status out of database
    // read temperature out of database
    // read priority out of database
    ...

    if (((temp_in < temp_min) && (temp_in < temp_out)) || ((temp_in >
temp_max) && -(temp_in >
temp_out)))
    {
        temp_alarm = true;
    }
    if (temp_in == temp_average)
    {
        temp_alarm = false;
    }
    ...

    // schreiben des Temperatur Alarm Statuses in die DB
    ...

    if (init && !win_open_bo && !state0)
    {
        init = false;
        state0 = true;
    }

    if (state0 && (priority==0) && !state11)
    {
        state0 = false;
        state11 = true;
    }

    // Transition 1.2
    if (state11 && (!wind_alarm && ((!autoMode_bo && manOpen_bo) ||
(autoMode_bo && humid_aIarm))) && !state12)
    {
        state11 = false;
        state12 = true;
    }
    ...

    // Assignments
    if(state12)
    {
        sw_send(DST_MULITCAST, "PUT_Win/Open");
        sleep(5);
    }

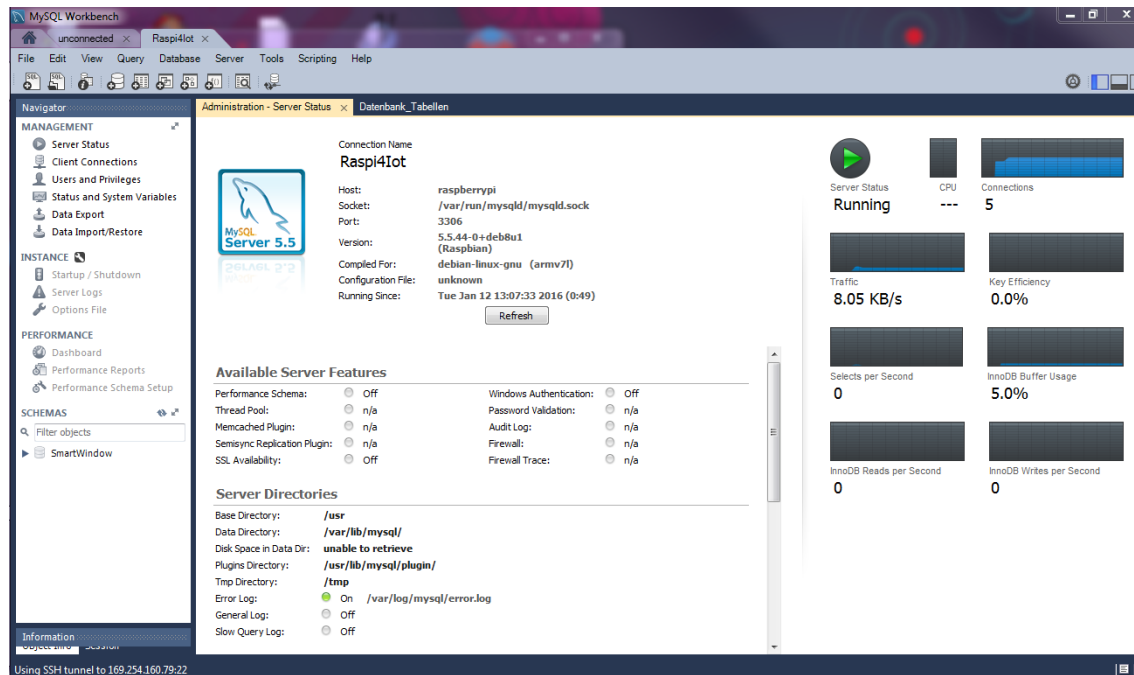
    if (state14 || init)
    {
        sw_send(DST_MULITCAST, "PUT_Win/Close");
        sleep(5);
    }
}
...

```

# Datenbank - MySQL

Smart  
Window

- Datenbank auf Raspberry
- MySQL Workbench



# Datenbank - MySQL

Smart  
Window

- Je eine Tabelle
  - Sensor
  - Limits
  - Befehle
  - Alarme

ID	Zeitstempel	Wert
1	14.01.2016 16:28:00	0
2	14.01.2016 16:29:00	0
3	14.01.2016 16:30:00	2

TEMP\_ALARMSTATE

ID	Zeitstempel	Wert
1	14.01.2016 16:28:00	22.00
2	14.01.2016 16:29:00	21.87
3	14.01.2016 16:30:00	21.42

TEMP\_IN

# Datenbank - MySQL

Smart  
Window

## MySQL Interface

```
bool write_in_db (MYSQL *mysql, const char *db,  
const char *table, const char *value)
```

```
int get_latest_value_int (MYSQL *mysql, const  
char *db, const char *table, bool *succeeded)
```

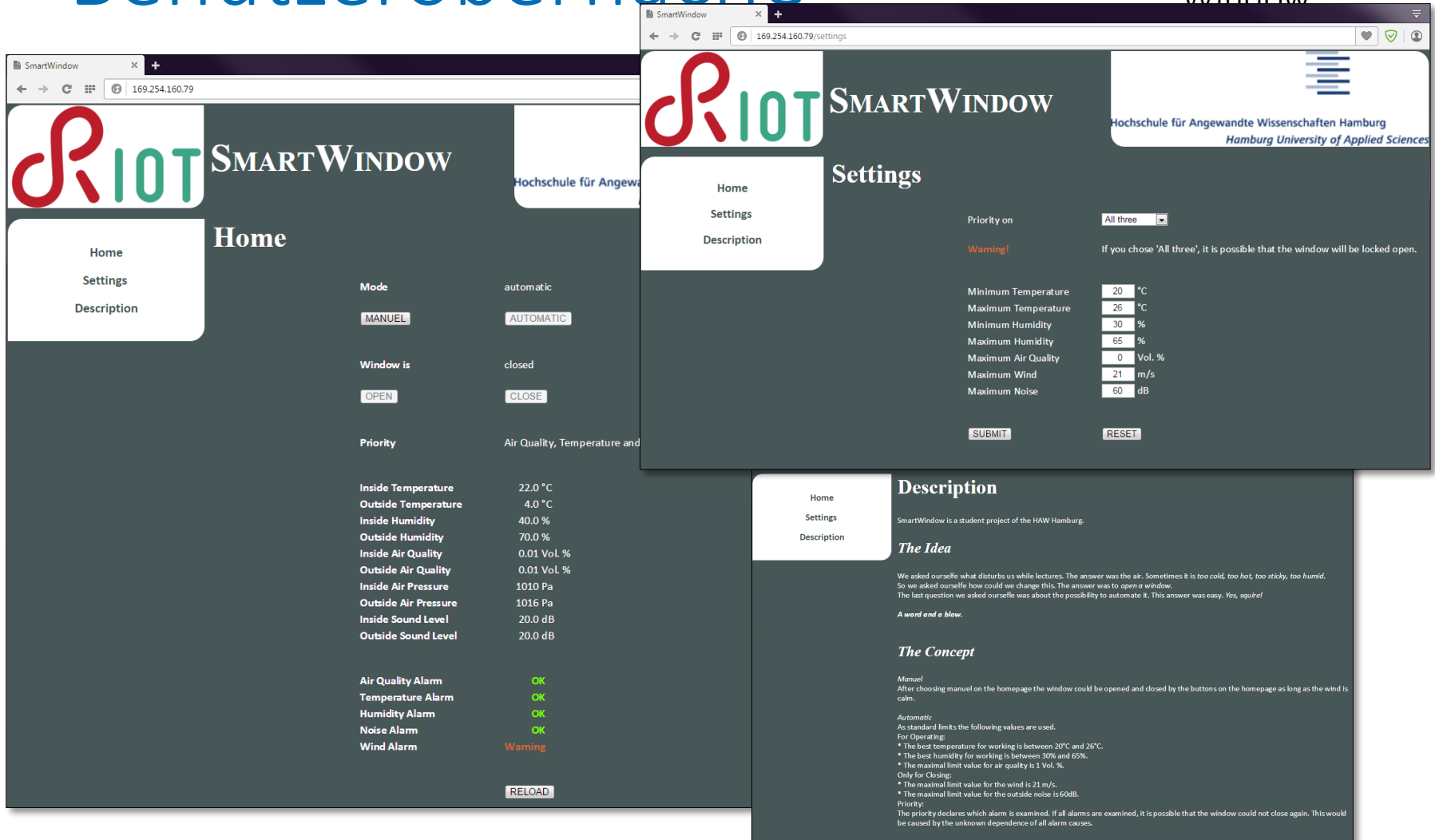
# Webserver

Smart  
Window

- Python Flask
- Anbindung an die Datenbank
- Übermittlung der Daten in Python an die HTML Dateien

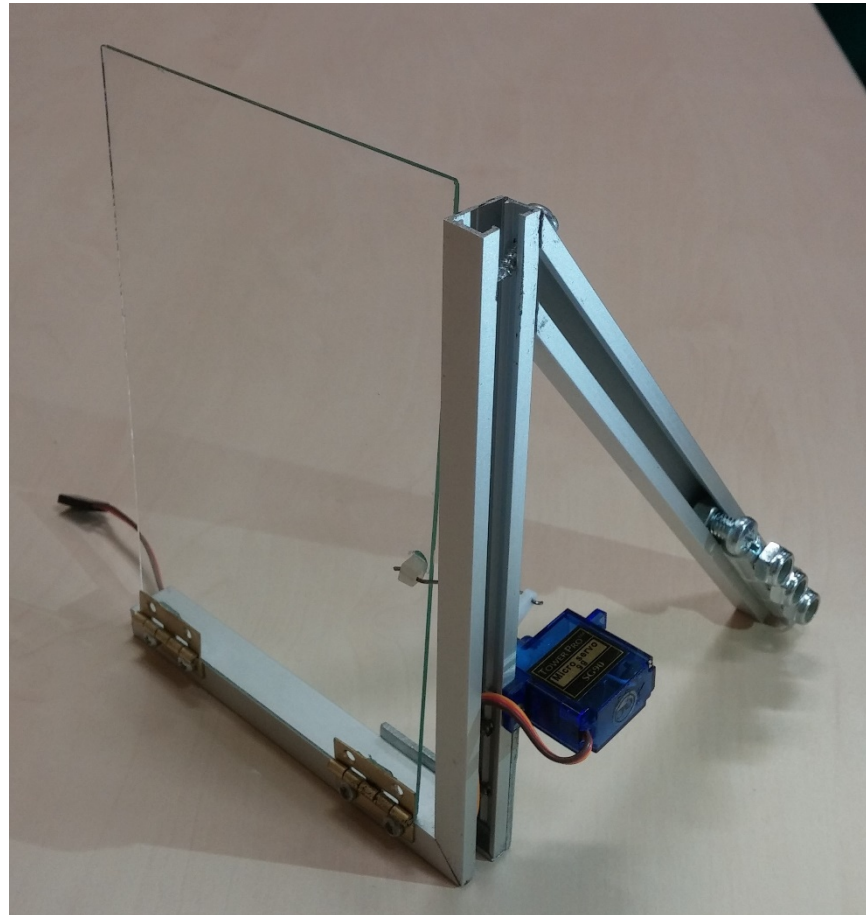
# Benutzeroberfläche

Smart Window



# Modell

Smart  
Window





Smart  
Window

# Live Vorführung

Smart  
Window

Fazit

# Feedback

Smart  
Window

- Was hat uns gut gefallen?
  - gute Atmosphäre
  - zu jeder Tageszeit hilfsbereite Dozenten
  - Projektcharakter (eigenständige Lösungsentwicklung)
  - Produktidee verwirklichung
  - Unterstützung von Informatikern
- Was könnte man verbessern?
  - Einweisung in github
  - Arbeiten mit Informatikstudenten

# Feedback

Smart  
Window

- Was konnten wir aus unserem Studium anwenden?
  - C, C++
  - Modellierung von Abläufen mit Petrinetzen
  - Datenbanken
  - Projektmanagement
- Was haben wir verbessert/ konnten wir neu erlernen?
  - Projektmanagement
  - Arbeiten im Team
  - Kommunikation
  - Tool: „github“
  - Kenntnisse im Umgang mit Linux

# Smart Window

---

Vielen Dank für die Aufmerksamkeit!